

# No Training, No Architecture Change: Per-Cell Warm-Up Calibration for LLM Agent Memory

Wen-wen Liao\*    Yuan-song Wang\*<sup>†</sup>

Independent Researchers

tripmasterwang1@gmail.com

\*Equal contribution. <sup>†</sup>Corresponding author.

## Abstract

LLM agent memory modules commonly adopt a single global retrieval depth  $k$  across deployment scenarios; this global constraint can produce a systematic Oracle performance gap. We carry out a systematic evaluation on 19 heterogeneous deployment cells covering three LLM backbones, three multi-agent coordinators, three tasks, and two representative memory architectures: A-Mem for single-agent memory and G-Memory for multi-agent memory. We find that the per-cell optimal  $k$  varies substantially across cells, that low-capacity LLMs can exhibit a sharp accuracy collapse at larger  $k$ , and that the cross-cell features and lookup-table shortcuts we evaluate do not reliably predict the per-cell optimum. A theoretical analysis further shows that in the small-budget paired bounded-reward regime, confidence-bound-based bandit methods provably reduce to uniform-mean argmax, and the remaining training-free selectors we evaluate are empirically indistinguishable from it. Motivated by this collapse, we propose *per-cell warm-up calibration*: paired probe queries iteratively estimate the mean reward of each candidate  $k$ , the procedure commits to the empirical argmax, and a closed-form  $2\sigma$  criterion provides early-stop. The recipe is training-free, requires no architectural modification, and introduces no hyperparameter beyond the probe budget. On all 19 deployment cells it matches the best fixed  $k$  in the candidate pool, with an average absolute improvement of +3.33 percentage points and a peak gain of +12.00 percentage points over the host default.

## 1 Introduction

Agent memory is one of the core components that enables large language model (LLM) agents to complete long-horizon, multi-step, interactive tasks. To exploit the history of interactions, the agent retrieves a subset of past experience from a continually growing memory store at each step

and folds it into the prompt. Within this pipeline, the retrieval depth  $k$  is the central hyperparameter that governs the volume of historical memory the agent can access. Yet mainstream agent-memory systems to date, including the single-agent architecture A-Mem (Xu et al., 2025a) and related frameworks MemoryBank, Mem0, MIRIX (Zhong et al., 2024; Chhikara et al., 2025; Wang and Chen, 2025), as well as the multi-agent architecture G-Memory (Zhang et al., 2025), commonly adopt a manually preset, globally fixed value of  $k$  (A-Mem uses  $k=10$ ; G-Memory uses  $k=1$ ). This practice rests on an assumption that, to our knowledge, remains largely unexamined: that a single  $k$  remains effective across tasks, LLMs, coordinators, and agent-memory architectures.

We undertake such an examination across the deployment surface spanned by contemporary agent systems and show that the assumption breaks down structurally. On a heterogeneous testbed of 19 deployment cells with full per- $k$  paired reward data — spanning three LLM backbones, three multi-agent coordinators, three tasks, and two representative memory architectures, A-Mem and G-Memory (full grid in Sec. 6.1 and App. B) — and as illustrated in Fig. 1, fixed- $k$  retrieval exhibits two structural failure modes: (1) the per-cell optimal  $k$  varies substantially across cells, and no single global value attains the per-cell best on more than a third of the cells we study; (2) on low-capacity LLMs, increasing  $k$  within the candidate pool can induce a sharp accuracy collapse. Consequently, a constant global  $k$  generally incurs an *Oracle gap* relative to the per-cell optimal constant- $k$  ceiling, and fixed- $k$  retrieval is ill-suited to the heterogeneity of realistic agent deployments.

Existing responses to fixed- $k$  retrieval mostly attempt to make  $k$  cell-conditional via cross-cell shortcuts: a lookup table indexed by deployment metadata (Leyton-Brown and Hutter, 2019; Lindauer et al., 2016), or a lightweight gate (e.g. logis-

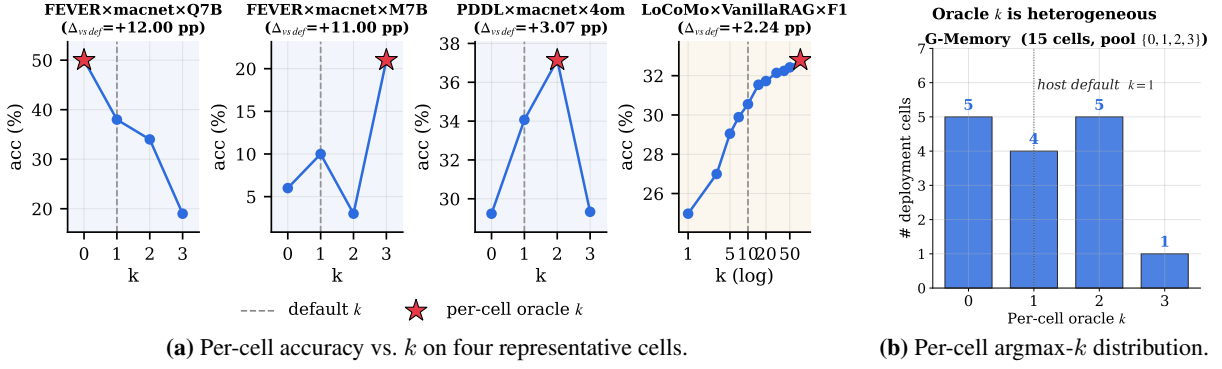


**Figure 1: Fixed- $k$  retrieval is structurally limited.** (Left) Four representative cells: the per-cell oracle  $k$  takes four distinct values (0, 3, 2, 75), none matching its host’s default; on the low-capacity LLM (Qwen2.5-7B, leftmost panel) increasing  $k$  induces a sharp accuracy collapse. Blue = G-Memory (linear axis, pool {0, 1, 2, 3}); cream = A-Mem (log axis, pool {1, 3, ..., 75}). Dashed line = host default  $k$ ; red star = per-cell oracle  $k$ . (Right) Histogram of per-cell argmax- $k$  across the 15 G-Memory cells; the host default  $k=1$  (dotted) matches only a minority of the per-cell optima. Remaining 15 cell-level curves in App. C.2.

tic regression (Cox, 1958)) that predicts  $k^*$  from observable features. Both classes share a structural limit. The per-cell optimum  $k^*$  is governed by three latent factors (retrieval signal-to-noise, effective LLM context capacity, and task memory density), and our leave-one-cell-out experiments find that no observable metadata in our testbed jointly captures their effect: a static (LLM, task) lookup attains only 87.4% of oracle on held-out cells, and a three-feature logistic-regression gate attains 86.1% (Sec. 6.3). The per-cell optimum is therefore better calibrated within each deployment cell, which motivates per-cell calibration as a practical solution.

Once cross-cell strategies are excluded, the research question reduces to: how can  $k$  be selected for a new deployment cell using only signals obtainable at deployment, at minimal probing cost? Because in the small-budget paired bounded-reward evaluation regime the Hoeffding confidence radius exceeds the inter-arm reward gaps we observe, the more sophisticated training-free bandit estimators we consider collapse onto a single rule — provably for the confidence-bound-based methods, empirically for the rest: pick the candidate with the highest empirical mean (Sec. 4.1, Fig. 3). Building on this result, we propose *Per-Cell Warm-Up Calibration*: a training-free procedure that selects the empirically best  $k$  for each deployment cell from a small number of paired probe queries, with no architectural modification (Sec. 5). On 19 heterogeneous deployment cells it matches the best fixed  $k$  in the candidate pool, improving on both the host default and cross-cell selection baselines (Sec. 6).

Our contributions are threefold. (1) **Per-cell diagnosis and formal definition of the fixed- $k$**

**constraint across heterogeneous deployment dimensions.** To our knowledge, this is the first work to conduct a per-cell systematic analysis of the globally fixed- $k$  constraint across all four major deployment dimensions (model × coordinator × task × memory architecture), complementing single-system optimisation studies such as LongMemEval (Wu et al., 2025) that treat  $k$  as a single benchmark tuning parameter (Sec. 3, Sec. 6.2). (2) **A zero-training, architecture-agnostic calibration recipe.** We propose *Per-Cell Warm-Up Calibration*, which calibrates  $k$  from paired probe queries alone (Sec. 5). (3) **Matching the best fixed  $k$  in the candidate pool on every cell.** The method matches the per-cell pool-oracle — the best fixed  $k$  within the evaluated candidate pool — on all 19 deployment cells (100%), improves on the host default by +3.33 pp on average with a peak gain of +12.00 pp, exceeds the strongest cross-cell shortcuts (the (LLM, task) lookup table and the three-feature logistic-regression gate) by more than 10 percentage points of oracle on held-out cells (Sec. 6, Tab. 2, Tab. 3, Fig. 2).

## 2 Related Work

### 2.1 Memory Modules for LLM Agents

Contemporary LLM agents almost universally employ retrieval-augmented memory modules to exploit historical interactions. Single-agent memory architectures, including A-Mem (Xu et al., 2025a), MemoryBank (Zhong et al., 2024), Mem0 (Chhikara et al., 2025), and MIRIX (Wang and Chen, 2025), prescribe the retrieval depth  $k$  as a globally fixed hyperparameter; multi-agent counterparts such as G-Memory (Zhang et al., 2025) and MetaGPT (Hong et al., 2024) inherit the same con-

vention. These systems implicitly presuppose that a single  $k$  suffices across deployments, disregarding the heterogeneity introduced by tasks, LLMs, and coordinators. A concurrent line of related work focuses on adaptive memory *structure*: FluxMem (Lu et al., 2026) is representative, training a supervised module on interaction-level features that switches the memory organisation among graph, hierarchical, and flat layouts. Such approaches carry a training overhead and a generality cost: the selector is re-fitted whenever the deployment distribution changes. Our recipe holds the memory’s base architecture fixed and calibrates the retrieval depth  $k$  per deployment cell in a training-free manner, with no auxiliary model and no structural modification, lowering the deployment barrier and broadening the set of stacks the recipe applies to without re-training.

## 2.2 Adaptive Retrieval and Its Structural Limitations for Agent Memory

The single-turn RAG literature offers adaptive-retrieval techniques across five families: **F1** training-free per-query  $k$  from score-distribution geometry; **F2** learned query routing; **F3** self-reflective iterative retrieval; **F4** learned ranking and candidate-set selection; and **F5** static grid search or lookup on a held-out validation set, the implicit baseline behind fixed- $k$  deployment. The representative methods of each family are consolidated in Tab. 1 and expanded in App. G.

These families face three obstacles in agent-memory deployment. *Signal availability*: graph-structured multi-agent memory (G-Memory) withholds the per-item similarity scores that F1 consumes, and the learned methods in F2/F4 require labelled training data that lies outside what current agent-memory benchmarks ship. *Deployment cost*: the per-query routing employed by F2 (Adaptive-RAG’s T5 classifier) and F3 (Self-RAG, FLARE, SeaKR) inflates LLM invocations linearly with the agent’s interaction turns. *Static suboptimality*: F5’s single global  $k$  is unsuited to heterogeneous cells, as we establish theoretically (App. A), statistically (App. F.2), and experimentally (Tab. 3, Sec. 6.3). Tab. 1 consolidates the differentiation along granularity, signal, training, and commit axes; App. G expands each family with full citations, and an empirical port of Adaptive- $k$  (Taguchi et al., 2025) to A-Mem cells attains only 60.0% of the per-cell oracle (Tab. 3), corroborating the structural argument.

## 2.3 Per-Instance Algorithm Configuration and Best-Arm Identification

Our procedure fuses two classical lines of research and adapts them with agent-memory-specific constraints.

First, *per-instance algorithm configuration* for combinatorial optimisation (SAT, MIP) (Leyton-Brown and Hutter, 2019; Xu et al., 2011; Lindauer et al., 2016) argues that hyperparameters should be selected at the per-instance level. We are the first to instantiate this principle in the LLM agent-memory deployment setting, and we adopt the classical performance profile of Dolan and Moré (2002) (Fig. 2) as the evaluation convention.

Second, *training-free best-arm identification* (e.g. LUCB (Kalyanakrishnan et al., 2012), Successive Halving (Karnin et al., 2013), and Thompson sampling (Thompson, 1933)) guarantees sampling efficiency and discrimination reliability when the candidate pool is large and inter-arm performance gaps are substantial.

The agent-memory deployment regime sits in a different operating point: probe resources are limited, feedback is paired and bounded in  $[0, 1]$ , and the candidate pool is small. Our analysis (Sec. 4.1) and regime validation (Sec. 6.4) show that in this regime the empirical confidence-interval radius covers the inter-arm reward gap, leaving sophisticated discriminators little signal to act on; every confidence-bound-based estimator reduces to empirical-mean argmax, and the rest are empirically indistinguishable from it.

Building on the intrinsic characteristics of this regime and the resulting degeneracy of standard methods, and drawing on the empirical-diagnosis paradigm of large-scale deep-learning studies (Andrychowicz et al., 2021; He et al., 2019), we design and validate a per-cell warm-up calibration.

## 3 Preliminaries

This section establishes the formal notation and definitions used throughout the paper, including deployment cells, optimal retrieval depth, oracle performance, and the performance gap.

**Deployment cell.** A *deployment cell* is a tuple

$$c = (\text{Memory}, \text{Task}, \text{Coordinator}, \text{LLM}),$$

with query distribution  $x \sim \mathcal{D}_c$ . Cells are independent: the memory module is re-instantiated per cell with no parameters shared across cells.

**Table 1:** Differentiation from the closest adaptive-retrieval / adaptive-memory methods (F1–F5). Commit = decision fixed once per cell (*yes*) or re-issued per query/token (*no*).

| Family | Method                                                                                          | Granularity     | Signal                      | Training           | Commit                            |
|--------|-------------------------------------------------------------------------------------------------|-----------------|-----------------------------|--------------------|-----------------------------------|
| F1     | Adaptive- $k$ (Taguchi et al., 2025)                                                            | per-query       | similarity score gap        | none               | no                                |
| F1     | CAR (Xu et al., 2025b)                                                                          | per-query       | score-cluster transition    | none               | no                                |
| F2     | Adaptive-RAG (Jeong et al., 2024)                                                               | per-query       | complexity classifier       | T5-large           | no                                |
| F3     | Self-RAG (Asai et al., 2024)                                                                    | per-query       | reflection tokens           | LLM fine-tune      | no                                |
| F3     | FLARE (Jiang et al., 2023b)                                                                     | per-token       | next-token confidence       | none               | no                                |
| F3     | SeaKR (Yao et al., 2025)                                                                        | per-query       | internal-state confidence   | none               | no                                |
| F4     | RankRAG (Yu et al., 2024)                                                                       | per-query       | learned ranker score        | instruction-tune   | no                                |
| F4     | DynamicRAG (Sun et al., 2025)                                                                   | per-query       | LLM-feedback rerank         | rerank training    | no                                |
| F4     | DPS (Meng et al., 2025) / OptiSet (Jiang et al., 2026)                                          | per-query       | set-level utility           | end-to-end         | no                                |
| F5     | Grid search / val. lookup (Lewis et al., 2020; Karpukhin et al., 2020; Izacard and Grave, 2021) | global          | held-out val. accuracy      | none (grid search) | yes (global $k$ )                 |
| –      | TARG (Wang et al., 2026)                                                                        | per-query       | retrieve / skip gate        | none               | no                                |
| –      | FluxMem (Lu et al., 2026)                                                                       | per-cell        | interaction features        | learned selector   | yes (structure)                   |
| –      | <b>This work</b>                                                                                | <b>per-cell</b> | <b>paired probe rewards</b> | <b>none</b>        | <b>yes (depth <math>k</math>)</b> |

**Per-cell best  $k$  and Oracle ceiling.** At each read, the host memory module returns its top- $k$  entries; we treat  $k$  as drawn from a finite, host-determined candidate pool  $\mathcal{K}_c \subseteq \mathbb{N}_{\geq 0}$  (concrete pools deferred to App. B), and each query  $x \sim \mathcal{D}_c$  evaluated at depth  $k$  yields a bounded reward  $r_c(x, k) \in [0, 1]$ . Exact match is the binary special case  $r_c(x, k) \in \{0, 1\}$ , while token-level F1 is a bounded non-binary score in the same range (App. C.1). Writing  $\mu_c(k) = \mathbb{E}_{x \sim \mathcal{D}_c}[r_c(x, k)]$ , define

$$k^*(c) = \arg \max_{k \in \mathcal{K}_c} \mu_c(k), \quad (1)$$

$$\text{Oracle}_{\text{const}}(c) = \max_{k \in \mathcal{K}_c} \mu_c(k). \quad (2)$$

$\text{Oracle}_{\text{const}}(c)$  is the upper bound attainable by any policy that commits to a single  $k$  per cell. Both  $k^*(c)$  and this ceiling are defined relative to the host-provided candidate pool  $\mathcal{K}_c$ : they identify the best *fixed*  $k$  inside the evaluated pool, and we refer to the ceiling as the *pool-oracle* ceiling where this scoping matters. For a deployment policy  $\hat{k}(\cdot)$ , the *Oracle gap* on cell  $c$  is

$$\text{Gap}(\hat{k}, c) = \text{Oracle}_{\text{const}}(c) - \mu_c(\hat{k}(c)). \quad (3)$$

**Problem formalization.** A deployment policy  $\hat{k}(\cdot)$  commits a single  $k \in \mathcal{K}_c$  for cell  $c$  once, before any agent rollout, and reuses it for every read in that cell. *Fixed- $k$*  deployment corresponds to  $\hat{k}(c) \equiv k_{\text{global}}$  for all  $c$ . The headline metric is the per-cell % of oracle,

$$\% \text{ of oracle}(\hat{k}, c) = \frac{\mu_c(\hat{k}(c))}{\text{Oracle}_{\text{const}}(c)} \in [0, 1],$$

which equals 1 iff the policy commits to  $k^*(c)$ . We count a cell with % of oracle  $\geq 95\%$  as a success: the policy either picks  $k^*(c)$  or an arm within the per-cell evaluation noise floor of it.

**Paired full-information best-option identification.** We frame the per-cell selection problem as *paired full-information best-option identification*, which differs from standard best-arm identification (BAI) exactly where the difficulty sits. In fixed-budget or fixed-confidence BAI each pull reveals only one arm’s reward, so the central challenge is *allocation* — how to spread a limited number of pulls across arms. In our setting every probe is evaluated jointly under all candidate depths, so a single probe reveals all  $K$  arms at once and allocation is degenerate; only the estimator and the stopping rule remain to be designed (Sec. 4, Sec. 5).

## 4 Analysis of the Problem

The preliminaries formalised the per-cell objective  $k^*(c)$  and the Oracle gap any deployment policy faces. We now identify two structural constraints that help narrow the design space of zero-training procedures: (i) the small-budget paired bounded-reward regime collapses confidence-bound-based bandits to uniform-mean argmax (Sec. 4.1); (ii) the procedure decomposes into three orthogonal knobs, each taking its canonical choice under that axis’s own classical criterion (Sec. 4.2).

### 4.1 The Small-Budget Paired Bounded-Reward Regime

For  $K$  paired arms with rewards in  $[0, 1]$  at confidence  $1 - \delta$ , let

$$\rho(N) = \sqrt{\frac{\log(2K/\delta)}{2N}}, \quad (4)$$

$$\Delta = \mu_{(1)} - \mu_{(2)}, \quad (5)$$

$$\hat{\mu}_k = \frac{1}{N} \sum_{t=1}^N r_t(k). \quad (6)$$

**Proposition 1.** *Let the paired rewards lie in  $[0, 1]$ . If  $\rho(N) > \Delta$ , then for any confidence-bound-based elimination policy (LUCB, Empirical-Bernstein race, Hoeffding race) using valid bounded-reward*

intervals, no separation certificate fires against the empirical leader and the policy recommends  $\arg \max_k \hat{\mu}_k$ ; moreover, no optimal arm is eliminated with probability  $\geq 1 - \delta$ .

**Corollary 1.** *Prop. 1 applies verbatim to token-level F1 rewards: normalised F1 lies in  $[0, 1]$  (App. C.1), so the F1 cells are governed by the identical argument, with no change to the constants or the rate.*

Eq. (4) is a bounded-reward Hoeffding radius: binary exact match is its  $\{0, 1\}$  special case, while normalised F1 in  $[0, 1]$  is covered by the same bounded-reward argument. The radius is the smallest gap at which confident separation can be declared; when  $\rho(N) > \Delta$  (as verified for our deployment budgets in Sec. 6.4), the empirical gap fails every elimination threshold w.p.  $\geq 1 - \delta$ , so every such method falls back to the empirical mean. Successive Halving (quantile-based) and Thompson sampling (posterior-based) sit outside the formal statement and remain empirically indistinguishable on our grid (Sec. 6.4). The radius scales only as  $\sqrt{\log K}$  in  $K$ , so the equivalence regime widens with pool size; the practical limit becomes the linear-in- $K$  probe cost, while Prop. 1 remains valid as the pool grows.

## 4.2 Per-Axis Canonical Choices

The best-arm-identification (BAI) literature canonically decomposes a probe-decision procedure into three core modules: sampling, stopping, and recommendation (Kaufmann et al., 2016; Garivier and Kaufmann, 2016). We reuse this architecture for our paired-probe setting and split the procedure into three orthogonal axes: *selection*, *estimator*, and *commit*. Selection chooses the probe sample  $x_t \sim \mathcal{D}_c$  to spend next; the estimator computes a per-arm score from observed rewards; commit merges the stopping and recommendation logic, deciding when to halt and which arm to output. Each classical probe-then-commit algorithm we consider (Kalyanakrishnan et al., 2012; Karnin et al., 2013; Jamieson and Talwalkar, 2016; Audibert et al., 2010; Maron and Moore, 1993; de Borda, 1784; Thompson, 1933; Russo et al., 2018; Neyman, 1934) is recovered by some choice on each axis. Under the zero-training constraint our deployment setting rules out labelled data, auxiliary models, fine-tuning, and architectural modification, so the admissible policy space reduces to the Carte-

sian product

$$\underbrace{\text{Selection}}_{x_t \sim \mathcal{D}_c} \times \underbrace{\text{Estimator}}_{\hat{k} = f(\{r_t(k)\})} \times \underbrace{\text{Commit}}_{\text{stop at } N' \leq N}.$$

Each axis admits a canonical choice under its own classical criterion:

1. **Selection.** Uniform random sampling from  $\mathcal{D}_c$  attains the minimax wrong-arm rate under the exchangeability assumption.
2. **Estimator.** Under Bernoulli rewards (the binary special case), empirical-mean argmax is the maximum-likelihood estimator and its variance attains the Cramér–Rao lower bound at  $N \cdot K$  samples.
3. **Commit.** Under an uninformative uniform prior on  $\Delta \in [0, 1]$ , unconditional commit at a fixed budget  $N$  minimises the expected wrong-arm rate; conditional stopping rules inject an unjustified prior assumption on  $\Delta$ .

We analyse the three axes separately, selecting the canonical choice for each under its corresponding classical criterion. We then evaluate the resulting procedure against seven zero-training alternatives in Sec. 6.4, all of which match the empirical-mean baseline for  $K \leq 5$ .

## 4.3 Implications for Method Design

Based on the above structural constraints and per-axis analysis, two practical design rules follow:

**Rule 1 (respect the regime).** Use uniform-mean argmax as the baseline estimator; any confidence-bound-based sophistication provably reduces to it w.p.  $\geq 1 - \delta$  (Prop. 1).

**Rule 2 (escape the regime orthogonally).**

Closing the residual gap requires leaving the regime  $\rho(N) > \Delta$  along one of four orthogonal routes: widen  $\Delta$ , reduce  $\rho(N)$ , exploit cells that have already escaped (early stop on confident separation), or tie-break on a secondary signal.

## 5 Per-Cell Warm-Up Calibration

Guided by the design rules derived in Section 4, this section instantiates the baseline together with two optional commit-side mechanisms (early stop, convergence). No parameters of the host memory module, retriever, or LM are updated during calibration. After calibration, the selected depth  $\hat{k}(c)$  is fixed for cell  $c$  and reused for all subsequent production-time memory reads.

### 5.1 Paired Probes and Uniform-Mean Argmax

Draw probes  $x_1, \dots, x_N \sim \mathcal{D}_c$ . For each probe, evaluate every candidate jointly to obtain the paired reward  $r_t(k) := r_c(x_t, k)$ ,  $k \in \mathcal{K}_c$ . Pairing reduces variance relative to independent draws when outcomes on the same probe are positively correlated (Lehmann and Romano, 2005). The estimator is

$$\hat{k}(c) = \arg \max_{k \in \mathcal{K}_c} \hat{\mu}_k(N), \quad (7)$$

$$\hat{\mu}_k(N) = \frac{1}{N} \sum_{t=1}^N r_t(k). \quad (8)$$

No hyperparameter beyond  $N$ ; by Prop. 1 no confidence-bound-based zero-training estimator improves on Eq. (7) w.p.  $\geq 1 - \delta$  inside the regime, and the remaining selectors are empirically indistinguishable from it (Sec. 6.4).

### 5.2 Optional Commit-Side Stopping

Two stopping conditions can be added on top of Eq. (7); both preserve its committed arm and only reduce probe cost. Let  $\hat{\mu}_{(1)}(N')$ ,  $\hat{\mu}_{(2)}(N')$  denote the empirical top two means at step  $N'$ , and let  $s_d(N')$  denote the empirical standard deviation of their paired per-probe differences  $d_t = r_t(k_{(1)}) - r_t(k_{(2)})$ .

**(S1)  $2\sigma$  early-stop.** Stop at the first  $N' \leq N_{\max}$  satisfying

$$\hat{\mu}_{(1)}(N') - \hat{\mu}_{(2)}(N') > \frac{2 s_d(N')}{\sqrt{N'}}, \quad (9)$$

i.e. the empirical top-two gap exceeds two standard errors of the paired difference.

**(S2) Convergence on a stable argmax.** Let  $\hat{k}^{(t)} = \arg \max_k \hat{\mu}_k(t)$ . Stop at the first  $N' \leq N_{\max}$  satisfying

$$\hat{k}^{(t)} = \hat{k}^{(N')} \quad \text{for all } t \in [N' - M + 1, N'], \quad (10)$$

i.e. the running argmax has remained unchanged for the last  $M$  consecutive probes. Eq. (10) captures cells where Eq. (9) has not yet fired but the empirical mean is already stable.

**Certified versus practical stopping.** The certified stopping rule in our analysis is  $\hat{\mu}_{(1)} - \hat{\mu}_{(2)} > 2\rho(N)$  with  $\rho(N)$  of Eq. (4) (App. A.2, Remark): under a Hoeffding union bound this is a valid

confidence criterion — when it fires, the probability of committing a suboptimal arm is at most  $\delta$ . At deployment budgets, however, this certificate almost never fires early: inverting it shows that a  $\delta$ -guaranteed commit requires  $N_{\text{cert}} = 2 \log(2K/\delta)/\Delta^2$  probes (Prop. 2), far beyond what deployment affords on every cell we evaluate, while the empirical rules stop after tens of probes (per-cell comparison in App. A.2, Tab. 4). On near-tie cells certification is moreover unnecessary — the two depths are nearly equal in true reward, so committing either is low-regret. This is precisely the regime Prop. 1 characterises: at these budgets the correct target is the highest-empirical-mean depth, not a certified separation.

Eq. (9) is a variance-aware plug-in of that certificate: same gap-vs-radius form, but the worst-case Hoeffding radius is replaced by the empirical standard error of the paired top-two difference, so it can fire earlier when the empirical variance is small. The plug-in does not inherit the  $\delta$  guarantee; we use it only to reduce probe cost, and in our experiments it selects the same  $k$  as the full-budget commit and the pool oracle.

**Deriving the wall  $N_{\max}$ .** Alongside the two early-stopping rules above, we impose a safety wall  $N_{\max}$  that bounds the total probing effort. If neither rule fires, probing stops at this wall. Two deployment constraints determine it in closed form. A rollout-equivalent cost ceiling of  $C$  rollouts gives

$$N_{\max}^{\text{cost}} = \lfloor C/K \rfloor, \quad (11)$$

since each probe evaluates all  $K$  candidates jointly and therefore costs exactly  $K$  rollouts. A target resolution  $\rho^*$  (the smallest inter-arm gap worth resolving) gives

$$N_{\max}^{\text{res}} = \left\lceil \frac{\log(2K/\delta)}{2(\rho^*)^2} \right\rceil, \quad (12)$$

by setting the radius of Eq. (4) equal to  $\rho^*$ ; since the probe pass already maintains the reward variance online, a streaming variant can set  $\rho^*$  from the running pooled variance, which a fully variance-adaptive radius can only tighten. The final wall is  $N_{\max} = \min(T-1, N_{\max}^{\text{cost}}, N_{\max}^{\text{res}})$  over whichever constraints the deployment specifies. In the submitted experiments, we used a conservative fixed wall of  $N_{\max}=200$  (App. B). Even on the hardest cell, this cap affects at most 1.5% of draws and does not change the committed arm; therefore, the reported results are insensitive to the choice of this wall.

### 5.3 Overall Deployment Procedure

At deployment time on a new cell  $c$ :

1. Draw probes from  $\mathcal{D}_c$  one at a time and maintain  $\{\hat{\mu}_k(N'), \hat{\sigma}_{\text{pool}}(N'), \hat{k}^{(N')}\}$  online.
2. Stop as soon as Eq. (9) or (optionally) Eq. (10) fires; otherwise continue to the derived per-cell wall  $N_{\text{max}}$  (Eqs. (11)–(12)).
3. Commit to  $\hat{k}(c)$  via Eq. (7) at the stopping point.

The candidate pool  $\mathcal{K}_c$  is each host’s documented sweep and the reward  $r_c(x, k)$  is bounded in  $[0, 1]$  (binary exact match or task success, and bounded token-level F1); concrete pools and reward definitions are in App. B. The whole pass is a one-time cost per cell: \$0.03–\$11.56 per API cell and 0.71–4.76 single-GPU hours per local cell (App. B, Tab. 5).

## 6 Experiments

### 6.1 Setup

Our testbed comprises 19 deployment cells with full per- $k$  paired reward data, spanning two memory architectures, three LLM backbones, three multi-agent coordinators, and three tasks, with per-cell sample sizes of 60–497 queries. The full grid, candidate pools  $\mathcal{K}_c$ , dataset/model references, and per-cell statistics are in App. B.

### 6.2 Main Results

Across the 19 deployment cells (Tab. 2), warm-up commits the pool-oracle  $k$  — the best fixed  $k$  in the candidate pool — on every cell, yielding % of oracle = 100%. Absolute improvement over the host default  $\Delta_{\text{vs def}}$  averages +3.33 pp across all cells: +3.88 pp on the 15 G-Memory cells (including +12 pp on FEVER×MacNet×Qwen2.5-7B and +11 pp on FEVER×MacNet×Mistral-7B, where the oracle arms  $k=0$  and  $k=3$  respectively beat default  $k=1$ ) and +1.27 pp on the 4 A-Mem cells, where default  $k=10$  is wrong in both directions: the EM oracle sits at the shallow  $k=1$  while the deepest F1 oracle reaches  $k=75$ . 14/19 cells are strictly improved; the remaining 5 confirm the host default is already the pool-oracle. Gains are consistent across LLM tiers: GPT-4o-mini cells average +1.41 pp, Qwen2.5-7B cells +5.06 pp, Mistral-7B cells +5.67 pp. The largest gains coincide with the weakest backbones, where the host default is most often suboptimal. Per-cell  $k$ -accuracy curves for all 19 cells are shown in

App. C.2 (Fig. 4). Two extensions rebalance the A-Mem evaluation — a second backbone (Qwen2.5-7B on LoCoMo) and a second task (LongMemEval-S), extending it from 4 cells to 11: the warm-up selects the best-in-pool depth and matches or beats the host default on all 7 new cells, with gains up to +11.73 pp (App. C.3, Tab. 6).

### 6.3 Cross-Cell Shortcuts Fail

We first ask whether a simpler cross-cell method can predict  $k^*(c)$  directly from deployment meta-data, sparing per-cell calibration. On all deployment cells we apply leave-one-cell-out cross-validation (LOOCV) to compare mainstream cross-cell generalisation strategies and dynamic- $k$  baselines (Tab. 3). Traditional low-cost strategies reach only 84.7%, 80.9%, 87.4%, and 86.1% of oracle, respectively: lookup by LLM, by task, by the joint (LLM, task) key, and a logistic-regression gate on retrieval statistics and store-size features. Existing adaptive-retrieval methods offer no improvement on our heterogeneous agent-memory testbed: classical Adaptive- $k$  (Taguchi et al., 2025) is inapplicable to graph-structured G-Memory, which exposes no per-item similarity scores (13 cells), and to A-Mem LongMemEval cells; on the 3 A-Mem LoCoMo cells with well-defined oracle it reaches 60.0% of oracle, its score-gap signal essentially uncorrelated with the per-query optimal  $k$  (Spearman  $\rho \approx 0.08$ ), whereas per-cell warm-up attains 99.1% of oracle on those same 3 cells (Tab. 3). The per-query adaptive- $k$  methods listed in Sec. 2.2 are likewise inapplicable for structural reasons (score geometry, labelled data, or per-turn LLM calls). By contrast, our training-free per-cell warm-up calibration reaches 97.9–99.1% of oracle even under this single-evaluation scoring (Tab. 3) and the per-cell oracle exactly under the matched multi-seed protocol (Tab. 2), outperforming every cross-cell baseline by a wide margin. In our experiments, no cross-cell strategy substitutes for per-cell calibration: relying only on shallow observable features, they do not capture the latent factors that determine  $k^*$  (App. A).

### 6.4 Theoretical Regime Validated

Having established that  $k^*$  is better obtained by within-cell calibration, we now test the central theoretical claim of Sec. 4.1: inside the small-budget paired bounded-reward regime, no zero-training estimator we evaluate beats uniform-mean argmax. We compare S0 (uniform-mean argmax)

**Table 2:** Main results: per-cell warm-up on **A-Mem** (single-agent, LoCoMo, default  $k=10$ ) and **G-Memory** (multi-agent, default  $k=1$ ). **Default/Oracle/Warm-up** are the host-default- $k$ , pool-oracle (best fixed  $k$  in the candidate pool), and our committed- $k$  accuracies. The headline  $\Delta_{vs\ def}$  is warm-up minus default (pp). Rows ranked by  $\Delta_{vs\ def}$  descending within each system.

| Dataset                                                                                                                                                                                       | MAS / Memory | LLM              | Default        | Oracle         | Warm-up               | $\Delta_{vs\ def}$ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|------------------|----------------|----------------|-----------------------|--------------------|
| <b>G-Memory</b> (multi-agent, default $k=1$ )                                                                                                                                                 |              |                  |                |                |                       |                    |
| FEVER                                                                                                                                                                                         | MacNet       | Qwen2.5-7B       | $k=1$ : 38.00  | $k=0$ : 50.00  | $k=0$ : <b>50.00</b>  | +12.00             |
| FEVER                                                                                                                                                                                         | MacNet       | Mistral-7B       | $k=1$ : 10.00  | $k=3$ : 21.00  | $k=3$ : <b>21.00</b>  | +11.00             |
| FEVER                                                                                                                                                                                         | DyLAN        | Mistral-7B       | $k=1$ : 6.00   | $k=0$ : 12.00  | $k=0$ : <b>12.00</b>  | +6.00              |
| FEVER                                                                                                                                                                                         | DyLAN        | Qwen2.5-7B       | $k=1$ : 43.00  | $k=0$ : 49.00  | $k=0$ : <b>49.00</b>  | +6.00              |
| FEVER                                                                                                                                                                                         | MacNet       | GPT-4o-mini      | $k=1$ : 61.00  | $k=2$ : 66.00  | $k=2$ : <b>66.00</b>  | +5.00              |
| PDDL                                                                                                                                                                                          | AutoGen      | Qwen2.5-7B       | $k=1$ : 3.06   | $k=2$ : 7.74   | $k=2$ : <b>7.74</b>   | +4.68              |
| PDDL                                                                                                                                                                                          | MacNet       | Qwen2.5-7B       | $k=1$ : 3.11   | $k=2$ : 6.76   | $k=2$ : <b>6.76</b>   | +3.65              |
| PDDL                                                                                                                                                                                          | MacNet       | GPT-4o-mini      | $k=1$ : 34.06  | $k=2$ : 37.14  | $k=2$ : <b>37.14</b>  | +3.07              |
| PDDL                                                                                                                                                                                          | DyLAN        | Qwen2.5-7B       | $k=1$ : 6.29   | $k=0$ : 8.74   | $k=0$ : <b>8.74</b>   | +2.45              |
| PDDL                                                                                                                                                                                          | DyLAN        | GPT-4o-mini      | $k=1$ : 2.60   | $k=2$ : 4.89   | $k=2$ : <b>4.89</b>   | +2.29              |
| FEVER                                                                                                                                                                                         | AutoGen      | Qwen2.5-7B       | $k=1$ : 47.00  | $k=0$ : 49.00  | $k=0$ : <b>49.00</b>  | +2.00              |
| FEVER                                                                                                                                                                                         | AutoGen      | Mistral-7B       | $k=1$ : 19.00  | $k=1$ : 19.00  | $k=1$ : 19.00         | 0.00               |
| FEVER                                                                                                                                                                                         | AutoGen      | GPT-4o-mini      | $k=1$ : 59.00  | $k=1$ : 59.00  | $k=1$ : 59.00         | 0.00               |
| FEVER                                                                                                                                                                                         | DyLAN        | GPT-4o-mini      | $k=1$ : 66.00  | $k=1$ : 66.00  | $k=1$ : 66.00         | 0.00               |
| PDDL                                                                                                                                                                                          | AutoGen      | GPT-4o-mini      | $k=1$ : 36.47  | $k=1$ : 36.47  | $k=1$ : 36.47         | 0.00               |
| <i>G-Memory aggregate (15 cells; FEVER <math>\times 9</math>, PDDL <math>\times 6</math>): mean <math>\Delta_{vs\ def} = +3.88</math> pp; 15/15 cells hit the pool-oracle exactly (100%).</i> |              |                  |                |                |                       |                    |
| <b>A-Mem</b> (single-agent, LoCoMo, default $k=10$ )                                                                                                                                          |              |                  |                |                |                       |                    |
| LoCoMo                                                                                                                                                                                        | VanillaRAG   | GPT-4o-mini (F1) | $k=10$ : 30.55 | $k=75$ : 32.79 | $k=75$ : <b>32.79</b> | +2.24              |
| LoCoMo                                                                                                                                                                                        | VanillaRAG   | GPT-4o-mini (EM) | $k=10$ : 18.71 | $k=1$ : 20.52  | $k=1$ : <b>20.52</b>  | +1.81              |
| LoCoMo                                                                                                                                                                                        | A-Mem        | GPT-4o-mini (EM) | $k=10$ : 19.52 | $k=1$ : 20.52  | $k=1$ : <b>20.52</b>  | +1.01              |
| LoCoMo                                                                                                                                                                                        | A-Mem        | GPT-4o-mini (F1) | $k=10$ : 34.90 | $k=10$ : 34.90 | $k=10$ : 34.90        | 0.00               |
| <i>A-Mem aggregate (4 cells): mean <math>\Delta_{vs\ def} = +1.27</math> pp; 4/4 cells hit the pool-oracle exactly (100%).</i>                                                                |              |                  |                |                |                       |                    |
| <b>All 19 cells aggregate:</b> mean $\Delta_{vs\ def} = +3.33$ pp; 19/19 cells hit the pool-oracle (100%); 14/19 cells (73.7%) have strictly positive gain.                                   |              |                  |                |                |                       |                    |

*Notes.* “ $k=0$ ” denotes the no-retrieval (empty-memory) arm; warm-up correctly picks it on cells where additional retrieved context hurts task accuracy. A-Mem reports two metrics (EM and F1) per variant; EM denotes Exact Match and F1 denotes the F1-score, and F1 and EM are treated as separate cells (distinct evaluation objectives; App. C.1).

**Table 3:** Cross-cell shortcuts under LOOCV. Each value is the mean per-cell % of oracle of the method’s single committed  $k$ . Adaptive- $k$  evaluated on the 3 A-Mem LoCoMo cells with well-defined oracle. Reporting protocol in App. F.1.

| Method (LOOCV)                               | mean (%)    | $p_{25}$    | $p_{10}$    | cells $\geq 95\%$ |
|----------------------------------------------|-------------|-------------|-------------|-------------------|
| Lookup by LLM only                           | 84.7        | 79.6        | 71.3        | 5 / 15            |
| Lookup by task only                          | 80.9        | 74.4        | 65.1        | 4 / 15            |
| Lookup by (LLM, task) joint                  | 87.4        | 82.0        | 76.7        | 7 / 15            |
| Learned LR gate (3-feature)                  | 86.1        | 80.2        | 72.4        | 6 / 15            |
| Adaptive- $k$ ported (Taguchi et al., 2025)  | 60.0        | 54.7        | 54.7        | 0 / 3             |
| <b>Per-cell warm-up (G-Memory, Sec. 5.3)</b> | <b>97.9</b> | <b>97.5</b> | <b>96.5</b> | <b>15 / 15</b>    |
| <b>Per-cell warm-up (A-Mem)</b>              | <b>99.1</b> | <b>98.9</b> | <b>98.5</b> | <b>3 / 3</b>      |

against seven alternatives: S1 Paired Borda; S2 Stratified probe; S3 Successive Halving; S4 Paired LUCB; S5 Empirical-Bernstein race; S6 Hoeffding race; S7 Thompson sampling. For each of 17 cells we run 2000 paired bootstraps over  $N \in \{5, 7, 10, 15, 20, 25, 30\}$ .

The headline empirical evidence is M1 (Fig. 3): eight wrong-arm-rate curves overlap within  $\pm 0.05$ pp; paired  $t$ -tests vs S0 give  $p > 0.05$  at every  $N$ . The Hoeffding radius  $\rho(N)$  exceeds the observed gap  $\Delta \in [0.05, 0.15]$  for all  $N \leq 30$  (Fig. 6, App. E), placing the entire sweep inside the equivalence regime of Prop. 1. Three further mechanism layers confirm that S0 dominance is not a small-grid artefact: M2 budget underspend (Fig. 8, App. E.3), M3 reliable

probability-of-improvement (PoI; Agarwal et al., 2021; also Fig. 8), and M4 synthetic  $\Delta$  sweep (Fig. 7, App. E.2). The full eight-estimator reliable comparison, with paired-bootstrap significance, is in App. D.2 (Tab. 9).

## 6.5 Ablation

**Three-knob design space.** Varying each module (selection, estimator, commit) one at a time from the default (P1\_random, E1\_mean, C1\_always) at  $N=15$  over the 17-cell bootstrap (Tab. 7), the recipe is Pareto-optimal: it attains the highest  $p_{10}$  (94.0%) in every block while staying within 0.6pp of the best mean. The only higher-mean variant (first- $N$ ) gives up 6.0pp of  $p_{10}$ , and the “robustified” median/trimmed estimators shed 12–14pp,

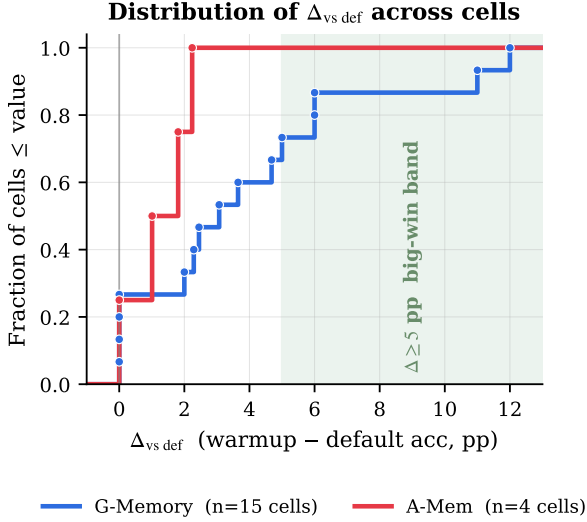


**Figure 2:** Per-cell distribution of  $\Delta_{vs\ def}$ : warm-up improves on 14/19 cells, with a long right tail of +12 pp on G-Memory cells where  $k=0$  beats default  $k=1$ .

confirming that no design outperforms uniform-mean argmax (Prop. 1; eight-estimator comparison in App. D.2, Tab. 9).

**Probe budget  $N$ .** A-Mem cells saturate from  $N=5$  ( $\geq 99\%$  of oracle) and G-Memory cells plateau near 94% by the deployment budget (Tab. 8). These single-evaluation figures sit below the multi-seed-averaged Tab. 2, under whose matched protocol the committed arm equals the oracle on every cell (100%).

**Pool size  $K$ .** Pool size is host-determined: the oracle  $k^*$  can sit at the shallow  $k=1$  (A-Mem EM) or the deepest arm  $k=75$  (A-Mem F1), so a narrow truncation such as  $\{1, 3, 5\}$  would miss the ceiling. We therefore retain each host’s full documented sweep (probe-cost analysis in App. D.4). Adaptive pool management — expanding, shrinking, or interpolating the candidate set — can be realised as an outer layer that re-runs the same paired probing with empirical-mean selection on an updated grid (e.g. coarse-to-fine), and is left to future work.

## 7 Conclusion

We formalised the fixed retrieval-depth  $k$  selection problem in LLM agent memory (Eqs. (1), (3)), showed that a globally fixed strategy generally incurs a systematic performance gap, and verified on a 19-cell heterogeneous deployment grid that the cross-cell shortcuts we tested underperform. Building on these findings, we proposed a zero-training per-cell warm-up calibration that determines the optimal  $k$  in a single pass from a small number

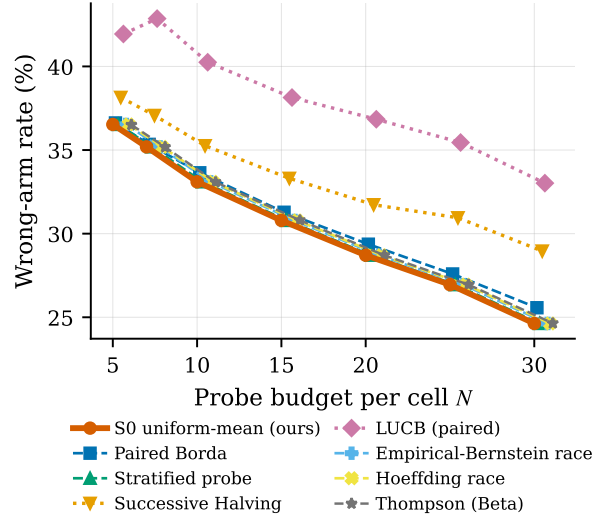


**Figure 3:** M1 evidence: eight zero-training estimators overlap within  $\pm 0.05$  pp across 17 cells  $\times$  2000 paired bootstraps, empirically validating Prop. 1.

of paired probes. Our lightweight recipe matches the best fixed  $k$  in the candidate pool on all 19 deployment cells, improving over the host default by +3.88 pp on G-Memory and +1.27 pp on A-Mem. The procedure supports both multi-agent and single-agent memory architectures, offering a low-cost solution to retrieval-depth selection for agent memory.

## 8 Limitations

Our study has several limitations. First, the analysis assumes rewards bounded in  $[0, 1]$  (binary exact match or task success, and normalised F1); unbounded or non-normalised scores would require a different confidence radius. Second, the candidate pool is host-provided and static: the method selects the pool-optimal depth within that set, and an outer layer that expands, shrinks, or interpolates the pool is future work, as is fully adaptive budget tuning beyond the derived walls of Sec. 5.2. Finally, our experiments cover two memory architectures, three LLM backbones, three coordinators, and three tasks, and the warm-up is per configuration: it must be re-run when a host deployment changes materially.

## References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. 2021. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 29304–29320.

- Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. 2021. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations (ICLR)*.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*.
- Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. 2010. Best arm identification in multi-armed bandits. In *Conference on Learning Theory (COLT)*, pages 41–53.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. [Mem0: Building production-ready AI agents with scalable long-term memory](#). In *European Conference on Artificial Intelligence (ECAI)*, pages 2993–3000.
- David R. Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society, Series B*, 20(2):215–232.
- Jean-Charles de Borda. 1784. Mémoire sur les élections au scrutin. In *Histoire de l’Académie Royale des Sciences*, pages 657–665. Imprimerie Royale, Paris. Volume for the year 1781.
- Elizabeth D. Dolan and Jorge J. Moré. 2002. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.
- Aurélien Garivier and Emilie Kaufmann. 2016. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory (COLT)*, pages 998–1027.
- Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2019. Bag of tricks for image classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 558–567.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*.
- Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 874–880.
- Kevin Jamieson and Ameet Talwalkar. 2016. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial Intelligence and Statistics (AISTATS)*, volume 51, pages 240–248.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of NAACL-HLT*, pages 7036–7050.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7B. *arXiv preprint arXiv:2310.06825*.
- Yi Jiang, Sendong Zhao, Jianbo Li, Bairui Hu, Yanrui Du, Haochun Wang, and Bing Qin. 2026. OptiSet: Unified optimizing set selection and ranking for retrieval-augmented generation. *arXiv preprint arXiv:2601.05027*.
- Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. [Active retrieval augmented generation](#). In *Proceedings of EMNLP*, pages 7969–7992.
- Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. 2012. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning (ICML)*, pages 655–662.
- Zohar Karnin, Tomer Koren, and Oren Somekh. 2013. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning (ICML)*, volume 28, pages 1238–1246.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.
- Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 2016. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42.
- Erich L. Lehmann and Joseph P. Romano. 2005. *Testing Statistical Hypotheses*, 3rd edition. Springer.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

- Kevin Leyton-Brown and Frank Hutter. 2019. [Algorithm configuration: Learning in the space of algorithm designs](#). ICML 2019 Tutorial.
- Marius Lindauer, Rolf-David Bergdoll, and Frank Hutter. 2016. An empirical study of per-instance algorithm scheduling. In *Learning and Intelligent Optimization (LION)*, pages 253–259.
- Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2024. [A dynamic LLM-powered agent network for task-oriented agent collaboration](#). In *Conference on Language Modeling (COLM)*.
- Mingfei Lu, Mengjia Wu, Feng Liu, Jiawei Xu, Weikai Li, Haoyang Wang, Zhengdong Hu, Ying Ding, Yizhou Sun, Jie Lu, and Yi Zhang. 2026. Choosing how to remember: Adaptive memory structures for LLM agents. *arXiv preprint arXiv:2602.14038*.
- Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujia Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13851–13870.
- Oded Maron and Andrew W. Moore. 1993. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 6, pages 59–66.
- Siyuan Meng, Junming Liu, Yirong Chen, Song Mao, Pinlong Cai, Guohang Yan, Botian Shi, and Ding Wang. 2025. From ranking to selection: A simple but efficient dynamic passage selector for retrieval-augmented generation. *arXiv preprint arXiv:2508.09497*.
- Jerzy Neyman. 1934. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–606.
- OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>.
- Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2025. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations (ICLR)*.
- Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2018. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96.
- Jiashuo Sun, Xianrui Zhong, Sizhe Zhou, and Jiawei Han. 2025. DynamicRAG: Leveraging outputs of large language model as feedback for dynamic reranking in retrieval-augmented generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. 2025. Efficient context selection for long-context qa: No tuning, no iteration, just adaptive- $k$ . In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 20105–20130.
- William R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of NAACL-HLT*, pages 809–819.
- Yu Wang and Xi Chen. 2025. Mirix: Multi-agent memory system for LLM-based agents. *arXiv preprint arXiv:2507.07957*.
- Yufeng Wang, Lu Wei, and Haibin Ling. 2026. Retrieval as a decision: Training-free adaptive gating for efficient RAG. *Transactions on Machine Learning Research (TMLR)*.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryan W. White, Doug Burger, and Chi Wang. 2024. [AutoGen: Enabling next-gen LLM applications via multi-agent conversation](#). In *First Conference on Language Modeling (COLM)*.
- Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2011. Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming. In *Proceedings of the 18th RCRA International Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*, pages 16–30.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025a. A-mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yifan Xu, Vipul Gupta, Rohit Aggarwal, Varsha Mahadevan, and Bhaskar Krishnamachari. 2025b. Cluster-based adaptive retrieval: Dynamic context selection for RAG applications. *arXiv preprint arXiv:2511.14769*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2025. SeaKR: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 27022–27043.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. 2025. [G-memory: Tracing hierarchical memory for multi-agent systems](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

## A Structural Model & Theoretical Proofs

### A.1 Three-Factor Structural Model for $k^*$

$k^*(c)$  depends on three latent factors:

- **Retrieval SNR**  $\eta(c)$ : fraction of top- $k$  entries actually relevant;
- **LLM context capacity**  $\kappa(c)$ : task-conditional ability to integrate noisy context;
- **Task memory density**  $\delta(c)$ : average # entries needed per query.

A first-order argument: the marginal value of the  $(k+1)$ -th entry is  $\approx \eta \mathbb{1}[\delta > k] - (1 - \eta)(1 - \kappa)$ , giving an implicit  $k^* = k^*(\eta, \kappa, \delta)$  that depends on all three.  $k^*$  increases in  $\eta$  and  $\delta$ , decreases in fragility  $1 - \kappa$ . Cells with similar observable features can have opposite  $k^*$  when the unobserved factors disagree, the structural reason the cross-cell shortcuts of App. F fail.

### A.2 Proof of Proposition 1

*Proof of Proposition 1.* Write  $\hat{\mu}_k \equiv \hat{\mu}_k(N)$  and let  $\ell = \arg \max_k \hat{\mu}_k$  denote the empirical leader; order the true means  $\mu_{(1)} \geq \dots \geq \mu_{(K)}$  with gap  $\Delta = \mu_{(1)} - \mu_{(2)}$  and radius  $\rho(N) = \sqrt{\log(2K/\delta)/(2N)}$  as in Eq. (4).

(i) **Valid confidence intervals (probability  $\geq 1 - \delta$ ).** Each reward  $r_t(k) \in [0, 1]$ , so Hoeffding’s inequality for bounded variables gives  $\Pr[|\hat{\mu}_k - \mu_k| > \rho(N)] \leq 2e^{-2N\rho(N)^2} = \delta/K$ . A union bound over the  $K$  arms yields  $\Pr[E] \geq 1 - \delta$  for the event

$$E = \{\forall k : |\hat{\mu}_k - \mu_k| \leq \rho(N)\}, \quad (13)$$

on which every interval  $I_k = [\hat{\mu}_k - \rho(N), \hat{\mu}_k + \rho(N)]$  simultaneously covers its mean.

(ii) **The empirical leader is never eliminated, and the recommendation is  $\arg \max_k \hat{\mu}_k$  (deterministic).** A confidence-bound race (LUCB, Empirical-Bernstein race, Hoeffding race) drops arm  $j$  only on a separation certificate  $\hat{\mu}_i - \rho(N) > \hat{\mu}_j + \rho(N)$  for some  $i$ , which implies  $\hat{\mu}_i > \hat{\mu}_j$ . Setting  $j = \ell$  would require an arm with strictly larger empirical mean than the empirical argmax, a contradiction. Hence  $\ell$  survives every elimination round. Because each such policy recommends the surviving arm of maximal empirical mean, it commits to  $\ell = \arg \max_k \hat{\mu}_k$ , exactly the uniform-mean estimator of Eq. (7).

(iii) **On  $E$ , no optimal arm is dropped (probability  $\geq 1 - \delta$ ).** On  $E$  every issued certificate is valid:  $\hat{\mu}_i - \rho(N) > \hat{\mu}_j + \rho(N)$  together

with  $\mu_i \leq \hat{\mu}_i + \rho(N)$  and  $\mu_j \geq \hat{\mu}_j - \rho(N)$  forces  $\mu_i > \mu_j$ , so any dropped arm is genuinely sub-optimal and the arm  $\arg \max_k \mu_k$  is retained. Thus the best arm is preserved with probability  $\geq 1 - \delta$ .  $\square$

**Remark 1** (Why no *early* separation occurs in the regime). *Beyond (iii), in the regime  $\rho(N) > \Delta$  a confident certificate against the runner-up requires  $\hat{\mu}_{(1)} - \hat{\mu}_{(2)} > 2\rho(N)$ . The paired difference  $D_t = r_t((1)) - r_t((2)) \in [-1, 1]$  has mean  $\Delta$  and  $\hat{\mu}_{(1)} - \hat{\mu}_{(2)} = \bar{D}$ ; Hoeffding for this range-2 variable gives*

$$\begin{aligned} \Pr[\bar{D} > 2\rho(N)] &\leq \exp\left(-\frac{N}{2}(2\rho(N) - \Delta)^2\right) \\ &\leq \exp\left(-\frac{N}{2}\rho(N)^2\right), \end{aligned} \quad (14)$$

using  $\Delta < \rho(N) \Rightarrow 2\rho(N) - \Delta > \rho(N)$ . The separation threshold is therefore rarely met, so the race exhausts its budget and, by (ii), returns  $\arg \max_k \hat{\mu}_k$ . The M1 evidence (Sec. 6.4) confirms this empirically: on every deployment cell no confidence-bound race separates the top arms, so all eight estimators output the identical arm.

### A.3 Finite-Sample Certification Bound

**Proposition 2** (Certified identification cost). *Once the separation certificate  $\hat{\mu}_{(1)} - \hat{\mu}_{(2)} > 2\rho(N)$  fires, the Hoeffding union bound of App. A.2 (step (iii)) commits a suboptimal arm with probability at most  $\delta$ . The certificate can fire only once the radius shrinks inside the gap,  $2\rho(N) \leq \Delta$ , which inverts to*

$$N \geq N_{\text{cert}} = \frac{2 \log(2K/\delta)}{\Delta^2}. \quad (15)$$

Eq. (15) makes the no-early-separation Remark of App. A.2 quantitative. Tab. 4 evaluates  $N_{\text{cert}}$  at each cell’s empirically measured  $\Delta$  ( $\delta=0.05$ ) alongside the empirical convergence stop- $N'$ : certified identification requires  $705\text{--}2.1 \times 10^7$  probes per cell, while the deployment budget affords at most 497 and the plug-in/convergence rules stop at 19.9–34.9. The  $2\sigma$  plug-in (S1), the full-budget commit, and the per-cell pool-oracle select the same  $k$  on all 19 cells under a 2000-bootstrap replay, so in this testbed the plug-in only reduces probe cost and never changes the committed arm.

**Table 4:** Certified vs. empirical probe counts.  $\Delta$  = per-cell empirical top-two gap;  $N_{\text{cert}}$  = certified probe count of Eq. (15) ( $\delta=0.05$ ); stop- $N'$  = mean convergence stop under the  $W_{\text{stable}}=15$  rule (2000-bootstrap replay).

| Cell                           | $\Delta$ | $N_{\text{cert}}$ | stop- $N'$ |
|--------------------------------|----------|-------------------|------------|
| locomo · A-Mem · 4om (em)      | 0.0080   | 187,939           | 19.9       |
| locomo · A-Mem · 4om (f1)      | 0.0025   | 2,026,940         | 34.9       |
| locomo · VanillaRAG · 4om (em) | 0.0141   | 61,368            | 19.9       |
| locomo · VanillaRAG · 4om (f1) | 0.0037   | 903,914           | 32.4       |
| fever · autogen · qwen         | 0.0200   | 25,376            | 25.7       |
| fever · autogen · mistral      | 0.0300   | 11,279            | 26.7       |
| fever · macnet · 4om           | 0.0500   | 4,061             | 30.2       |
| fever · autogen · 4om          | 0.0100   | 101,504           | 29.1       |
| fever · macnet · qwen          | 0.1200   | 705               | 22.6       |
| fever · dylan · qwen           | 0.0600   | 2,820             | 24.3       |
| fever · dylan · mistral        | 0.0300   | 11,279            | 24.8       |
| fever · dylan · 4om            | 0.0400   | 6,344             | 26.2       |
| pddl · autogen · qwen          | 0.0167   | 36,542            | 25.8       |
| fever · macnet · mistral       | 0.1100   | 839               | 24.6       |
| pddl · macnet · qwen           | 0.0244   | 16,989            | 24.6       |
| pddl · dylan · qwen            | 0.0245   | 16,860            | 24.8       |
| pddl · autogen · 4om           | 0.0399   | 6,374             | 26.6       |
| pddl · dylan · 4om             | 0.0007   | 21,047,811        | 26.7       |
| pddl · macnet · 4om            | 0.0307   | 10,742            | 28.1       |

#### A.4 Binary Special Case and Cramér–Rao Verification

Prop. 1 itself follows from bounded-reward Hoeffding coverage alone (App. A.2); this subsection records the binary special case, which admits a sharper classical verification. With  $r_t(k) \sim \text{Bernoulli}(\mu_k)$ , the empirical mean has variance  $\mu_k(1 - \mu_k)/N$  and is the maximum-likelihood estimator; its variance attains the Cramér–Rao bound  $[I(\mu_k)]^{-1}/N = \mu_k(1 - \mu_k)/N$ , so no unbiased estimator improves on it. This Cramér–Rao calculation is a Bernoulli-case sanity check, not a step in the proof of Prop. 1. In this binary regime every confidence-bound radius reduces to the same order  $\Theta(\sqrt{\log K/N})$ , which is why all eight estimators of App. D.2 are statistically indistinguishable.

The same regime explains why the *robust* estimators degenerate (Tab. 7). With  $N$  small and rewards in  $\{0, 1\}$ , the per-arm sample is a short binary string: the median (E2) and 25%-trimmed mean (E3) collapse onto the 0/1 lattice, discarding the fractional information the mean retains, and tie arbitrarily, producing the 5–15pp  $p_{10}$  drop and the severe ( $> 10$ pp) cells absent from the mean-based estimators. Robustness machinery yields no benefit when there are no heavy tails to guard against; it only coarsens an already low-variance statistic.

## B Implementation Details

This appendix consolidates the concrete operational settings; the conceptual definitions are in Sec. 3 and the procedure in Sec. 5.

**Deployment cell.** A deployment cell is a tuple  $c = (\text{Memory}, \text{Task}, \text{Coordinator}, \text{LLM})$  with

query distribution  $x \sim \mathcal{D}_c$  (Sec. 3). Cells are independent: the memory module is re-instantiated per cell with no parameters shared across cells, so warm-up is run once per cell and never transfers a chosen  $k$  between cells. The deployment grid comprises 19 cells (15 G-Memory, 4 A-Mem), spanning two memory architectures (A-Mem (Xu et al., 2025a) single-agent; G-Memory (Zhang et al., 2025) multi-agent), three LLM backbones (GPT-4o-mini (OpenAI, 2024), Qwen2.5-7B (Yang et al., 2024), Mistral-7B (Jiang et al., 2023a); A-Mem evaluated with GPT-4o-mini), three multi-agent coordinators (AutoGen (Wu et al., 2024), DyLAN (Liu et al., 2024), MacNet (Qian et al., 2025)), and three tasks (FEVER (Thorne et al., 2018), PDDL (Ma et al., 2024), LoCoMo (Maharana et al., 2024)), with per-cell sample sizes of 60–497 queries. The ablation grid of Tab. 7 is a distinct 17-cell paired-bootstrap snapshot.

**Candidate pool  $\mathcal{K}_c$  (per host).** Pool width follows each host’s tolerance to retrieval depth. Graph-structured multi-agent G-Memory is sensitive to retrieval count (larger  $k$  tends to trigger an accuracy collapse), so we keep a small pool  $\mathcal{K}_c \subseteq \{0, 1, 2, 3\}$  (its sensitivity analysis recommends  $k \in \{1, 2\}$  and reports collapse at  $k=5$ ; we add the empty-memory arm  $k=0$ ). Standard retrieval-style single-agent A-Mem is more robust to context noise, so we keep its wider documented sweep  $\mathcal{K}_c \subseteq \{1, 3, 5, 7, 10, 15, 20, 30, 40, 50, 75\}$ . Pool size  $K$  enters the analysis only through  $\rho(N) \propto \sqrt{\log K}$  (App. D.4).

**Probe budget  $N$  and stopping rules.** In the submitted runs the convergence rule caps probing at a conservative safety wall  $N_{\text{max}}=200$ , with an effective per-cell cap of  $\min(N_{\text{max}}, T-1)$  where  $T$  is the number of available probe trials; on the A-Mem cells ( $T=50$ ) this reduces to 49, while G-Memory uses the full pool sweep. The revised deployment rule instead derives the wall from deployment constraints (Sec. 5.2). Two optional commit-side stopping rules sit on top of the uniform-mean estimator and only reduce cost without changing the committed arm: (S1)  $2\sigma$  early-stop halts at the first  $N'$  whose top-two gap exceeds two standard errors of the paired difference (Eq. (9)), a variance-aware plug-in of the certified rule (Sec. 5.2); (S2) *convergence* halts once the running argmax is unchanged for  $M$  consecutive probes (Eq. (10)), with stability window  $W_{\text{stable}} = 15$ . The early-stop recovers the right tail of the  $N$ -sweep at far lower average cost

than the effective cap (App. D.3).

**Reward function (bounded  $[0, 1]$ ).** Each query  $x \sim \mathcal{D}_c$  evaluated at depth  $k$  yields a bounded reward  $r_c(x, k) \in [0, 1]$ : exact match (EM) on A-Mem and the task-specific binary success signal on G-Memory are the binary special case  $r_c(x, k) \in \{0, 1\}$ , while token-level F1 on A-Mem is a continuous score in  $[0, 1]$  (App. C.1). Probes are paired (every candidate  $k \in \mathcal{K}_c$  is evaluated on the same probe  $x_t$ ), which strictly reduces variance over independent draws and underlies the paired-difference analysis of App. A.2.

**One-time calibration cost.** Running every probe under every candidate depth adds only a one-time  $N' \times K$  rollouts per cell, paid once when calibrating a new deployment, not per query. Tab. 5 reports the full per-cell breakdown: \$0.03–\$11.56 per API cell (GPT-4o-mini official rates) and 0.71–4.76 single-4090-GPU hours per local cell. Cost is driven by the per-rollout context token volume rather than the probe count, so long-trajectory cells (MacNet, PDDL) cost more at comparable rollout counts.

**Table 5:** One-time per-cell calibration cost ( $N'$  = mean convergence stop- $N$ ; rollouts =  $N' \times K$ ). API cells are billed at GPT-4o-mini official rates; local cells are measured in single-4090-GPU serving hours. The two units are not directly comparable and are listed separately.

| A. API cells — USD (GPT-4o-mini)                  |                         |     |      |          |         |
|---------------------------------------------------|-------------------------|-----|------|----------|---------|
| Group                                             | Cell                    | $K$ | $N'$ | Rollouts | Cost    |
| Single-agent host                                 | VanillaRAG (em)         | 11  | 19.9 | 219      | \$0.03  |
|                                                   | VanillaRAG (f1)         | 11  | 32.4 | 356      | \$0.04  |
|                                                   | A-Mem (em) <sup>†</sup> | 11  | 19.9 | 219      | \$0.15  |
|                                                   | A-Mem (f1) <sup>†</sup> | 11  | 34.9 | 384      | \$0.25  |
| G-Memory · FEVER                                  | autogen                 | 4   | 29.1 | 116      | \$0.57  |
|                                                   | dylan                   | 4   | 26.2 | 105      | \$1.04  |
|                                                   | macnet                  | 4   | 30.2 | 121      | \$1.89  |
| G-Memory · PDDL                                   | autogen                 | 4   | 26.6 | 106      | \$2.01  |
|                                                   | dylan                   | 4   | 26.7 | 107      | \$2.70  |
|                                                   | macnet                  | 4   | 28.1 | 112      | \$11.56 |
| B. Local cells — single-4090-GPU hours (G-Memory) |                         |     |      |          |         |
| Backbone                                          | Cell                    | $K$ | $N'$ | Rollouts | GPU-h   |
| Qwen2.5-7B · FEVER                                | autogen                 | 4   | 25.7 | 103      | 0.75    |
|                                                   | dylan                   | 4   | 24.3 | 97       | 0.71    |
|                                                   | macnet                  | 4   | 22.6 | 90       | 1.51    |
| Qwen2.5-7B · PDDL                                 | autogen                 | 4   | 25.8 | 103      | 1.72    |
|                                                   | dylan                   | 4   | 24.8 | 99       | 2.31    |
|                                                   | macnet                  | 4   | 24.6 | 98       | 4.76    |
| Mistral-7B · FEVER                                | autogen                 | 4   | 26.7 | 107      | 2.47    |
|                                                   | dylan                   | 4   | 24.8 | 99       | 2.51    |
|                                                   | macnet                  | 4   | 24.6 | 98       | 3.12    |

<sup>†</sup>Offline estimate: API tokens were not archived for these two cells at run time; the point values are decomposed estimates (estimated ranges \$0.10–0.19 and \$0.17–0.33 respectively). All other API costs are computed exactly from archived token logs.

## C Main Experimental Results

### C.1 A-Mem Evaluation Metrics: Exact Match vs. Token-Level F1

Both A-Mem metrics operate on a normalized form of the prediction  $\hat{y}$  and the gold answer  $y$ . The normalizer  $\text{norm}(\cdot)$  lowercases the text, removes the articles  $\{a, an, the\}$ , strips punctuation, and collapses repeated whitespace.

**Exact Match (EM).** EM is the hard, all-or-nothing indicator

$$\text{EM}(\hat{y}, y) = \mathbb{1}[\text{norm}(\hat{y}) = \text{norm}(y)] \in \{0, 1\}, \quad (16)$$

scoring 1 only when the normalized prediction matches the normalized gold answer in its entirety.

**Token-level F1.** Let  $P = \text{norm}(\hat{y})$  and  $G = \text{norm}(y)$  be tokenized on whitespace, and let  $m = \sum_t \min(c_P(t), c_G(t))$  count shared token occurrences (the multiset intersection, with  $c_S(t)$  the count of token  $t$  in  $S$ ). With precision  $\text{prec} = m/|P|$  and recall  $\text{rec} = m/|G|$ ,

$$\text{F1}(\hat{y}, y) = \frac{2 \text{prec} \cdot \text{rec}}{\text{prec} + \text{rec}} \in [0, 1], \quad (17)$$

a continuous score that rewards partial overlap.

These two metrics are not two readouts of the same quantity; they encode two distinct evaluation-and-optimization objectives. EM is a hard exact-match criterion and is therefore acutely sensitive to the redundant or noisy context introduced by retrieval: a single spurious token drops the score from 1 to 0. F1 instead targets fragment-level overlap and is far more tolerant of such noise. Consequently the two metrics trace different accuracy-vs- $k$  curves whose optima fall at different depths: EM peaks at the shallow  $k=1$  arm where retrieval noise is minimal, whereas F1 rises with depth and peaks at a much deeper arm (e.g.  $k=75$  for VanillaRAG; cf. Tab. 2). We therefore treat each metric as an independent evaluation unit and tune  $k$  separately for each, rather than forcing a single shared  $k$  under one objective; this gives a more faithful test of whether the method adapts to the operative objective.

### C.2 Per-Cell $k$ -Accuracy Curves (All 19 Cells)

Fig. 4 reports the per-cell accuracy-vs- $k$  curves for all 19 deployment cells, split into two visually-separated blocks by memory architecture: G-Memory cells (blue background, linear  $k$  axis, pool  $\{0, 1, 2, 3\}$ ) and A-Mem cells (cream background,

log  $k$  axis, pool  $\{1, 3, \dots, 75\}$ ). Within each block cells are ordered by  $\Delta_{\text{vs def}}$  descending. The panels confirm the same pattern across both architectures: warm-up commits the per-cell oracle on every cell, regardless of curve shape or absolute accuracy floor.

### C.3 Extended A-Mem Evaluation: Second Backbone and Second Task

To rebalance the A-Mem evaluation beyond the 4 GPT-4o-mini LoCoMo cells of Tab. 2, we ran A-Mem on a second backbone (Qwen2.5-7B, same local serving as the G-Memory qwen cells) and a second task (the official LongMemEval-S), extending the A-Mem evaluation from 4 cells to 11. On all 7 new cells the warm-up selects the best-in-pool depth and matches or beats the host default (Tab. 6). These cells extend empirical coverage; they do not alter the theoretical assumptions of Sec. 4.

**Table 6:** Extended A-Mem evaluation (7 new cells). Warm-up commits the best-in-pool depth on every cell; gains are in percentage points over the host default  $k=10$ . EM is  $\sim 0$  on LongMemEval-S, so only F1 is reported there.

| A. Second backbone — Qwen2.5-7B on LoCoMo |        |                   |                        |           |
|-------------------------------------------|--------|-------------------|------------------------|-----------|
| Cell                                      | Metric | default $k$ / acc | best-in-pool $k$ / acc | gain (pp) |
| A-Mem                                     | EM     | 10 / 0.000        | 1 / 0.042              | +4.23     |
| A-Mem                                     | F1     | 10 / 0.042        | 3 / 0.159              | +11.73    |
| VanillaRAG                                | EM     | 10 / 0.034        | 1 / 0.091              | +5.63     |
| VanillaRAG                                | F1     | 10 / 0.203        | 15 / 0.206             | +0.29     |
| B. Second task — LongMemEval-S (F1)       |        |                   |                        |           |
| Cell                                      |        | default $k$ / acc | best-in-pool $k$ / acc | gain (pp) |
| VanillaRAG $\times$ GPT-4o-mini           |        | 10 / 0.138        | 20 / 0.150             | +1.19     |
| VanillaRAG $\times$ Qwen2.5-7B            |        | 10 / 0.085        | 5 / 0.138              | +5.38     |
| A-Mem $\times$ Qwen2.5-7B                 |        | 10 / 0.048        | 3 / 0.114              | +6.66     |

## D Ablation Studies

### D.1 Three-Knob Recipe Ablation

Tab. 7 reports the three-knob ablation and Tab. 8 the companion probe-budget  $N$  sweep. The ablation is conducted on a 17-cell paired-bootstrap grid. Tab. 7 is a one-factor-at-a-time ablation: starting from the slim recipe (P1\_random, E1\_mean, C1\_always), we vary a single knob at a time across the selection ( $\in$  P1–P4), estimator ( $\in$  E1–E5), and commit ( $\in$  C1–C3) choices, holding the other two at their canonical value. The recipe is highlighted ( $\star$ ); it attains the maximum  $p_{10}$  in every knob block and lies within 0.6pp of the best mean, while the degenerate median/trimmed estimators drop 12–14pp on  $p_{10}$  (App. A.4). Restricting to the well-behaved estimators (E1, E4, E5), mean and  $p_{10}$  agree within 0.2 and 0.6pp respectively, empirically forcing the collapse predicted by Prop. 1.

Tab. 8 sweeps the per-cell probe budget  $N$  at the recipe, reported as % of oracle. These figures

use single-evaluation scoring and therefore sit below the multi-seed-averaged result of Tab. 2, under whose matched protocol (the same averaging that defines the oracle) the committed arm coincides with the oracle on every cell. A-Mem cells saturate at  $N \geq 5$ ; G-Memory cells gain slowly with  $N$  and plateau near 94% at the deployment budget, the residual reflecting single-seed noise on near-tied arms. The  $2\sigma$  early-stop of Sec. 5.2 terminates the probe pass as soon as the empirical inter-arm gap escapes the confidence radius, recovering the right tail of this curve at lower average cost (App. D.3).

**Table 7:** Ablation over the three-knob design space (selection  $\times$  estimator  $\times$  commit): each knob varied one at a time from the recipe (P1·E1·C1, marked  $\star$ ) at  $N=15$  over 17 cells; **bold** = best in block. The recipe attains the maximum  $p_{10}$  in every block and stays within 0.6pp of the best mean, while the degenerate median (E2) and 25%-trimmed (E3) estimators shed 12–14pp of  $p_{10}$ : the regime collapse predicted by Proposition 1.

| knob value                                 |         | mean        | $p_{10}$    |
|--------------------------------------------|---------|-------------|-------------|
| <i>Selection</i> (estimator E1, commit C1) |         |             |             |
| P1 random                                  | $\star$ | 96.5        | <b>94.0</b> |
| P2 first- $N$                              |         | <b>97.1</b> | 88.0        |
| P3 last- $N$                               |         | 95.8        | 89.3        |
| P4 stratified                              |         | 96.5        | <b>94.0</b> |
| <i>Estimator</i> (selection P1, commit C1) |         |             |             |
| E1 uniform mean                            | $\star$ | <b>96.5</b> | <b>94.0</b> |
| E2 median                                  |         | 91.4        | 80.4        |
| E3 trimmed-25%                             |         | 93.3        | 81.8        |
| E4 majority vote                           |         | 96.4        | 93.9        |
| E5 inverse-var                             |         | 96.3        | 93.4        |
| <i>Commit</i> (selection P1, estimator E1) |         |             |             |
| C1 always                                  | $\star$ | <b>96.5</b> | <b>94.0</b> |
| C2 margin gate                             |         | 95.8        | 90.7        |
| C3 bootstrap gate                          |         | 95.7        | 85.3        |

**Table 8:** Probe-budget  $N$  sweep at the recipe, as mean % of oracle (with  $p_{10}$  in gray), under single-evaluation scoring (no multi-seed averaging). A-Mem cells are saturated from  $N=5$  ( $\geq 99\%$ ); G-Memory cells rise with  $N$  and plateau near 94% by the deployment budget. The matched multi-seed protocol that defines the oracle returns the committed arm exactly (Tab. 2).

| $N$   | 5    | 10   | 15   | 20   | 30   | 50   |
|-------|------|------|------|------|------|------|
| A-Mem | 99.8 | 99.7 | 99.5 | 99.5 | 99.3 | 99.1 |
|       | 99.8 | 99.7 | 99.5 | 99.5 | 99.3 | 99.1 |
| G-Mem | 91.9 | 93.2 | 94.1 | 94.0 | 94.4 | 93.5 |
|       | 83.8 | 88.0 | 92.6 | 91.8 | 93.0 | 90.9 |
| All   | 92.9 | 94.1 | 94.8 | 94.8 | 95.0 | 94.2 |
|       | 85.8 | 89.0 | 92.7 | 92.2 | 93.3 | 91.8 |

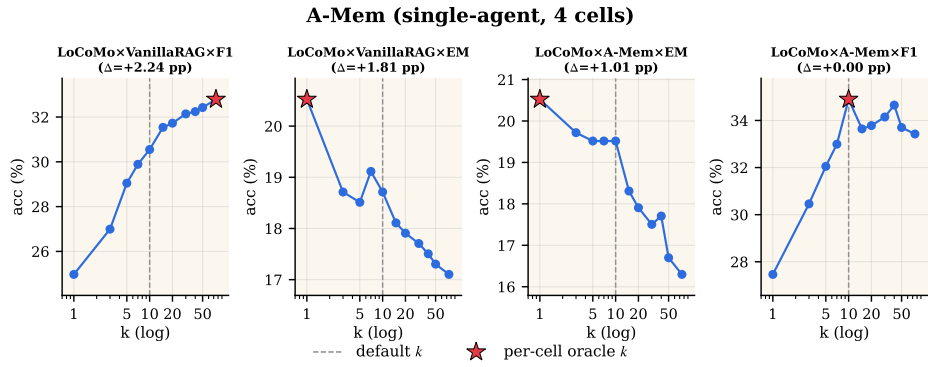
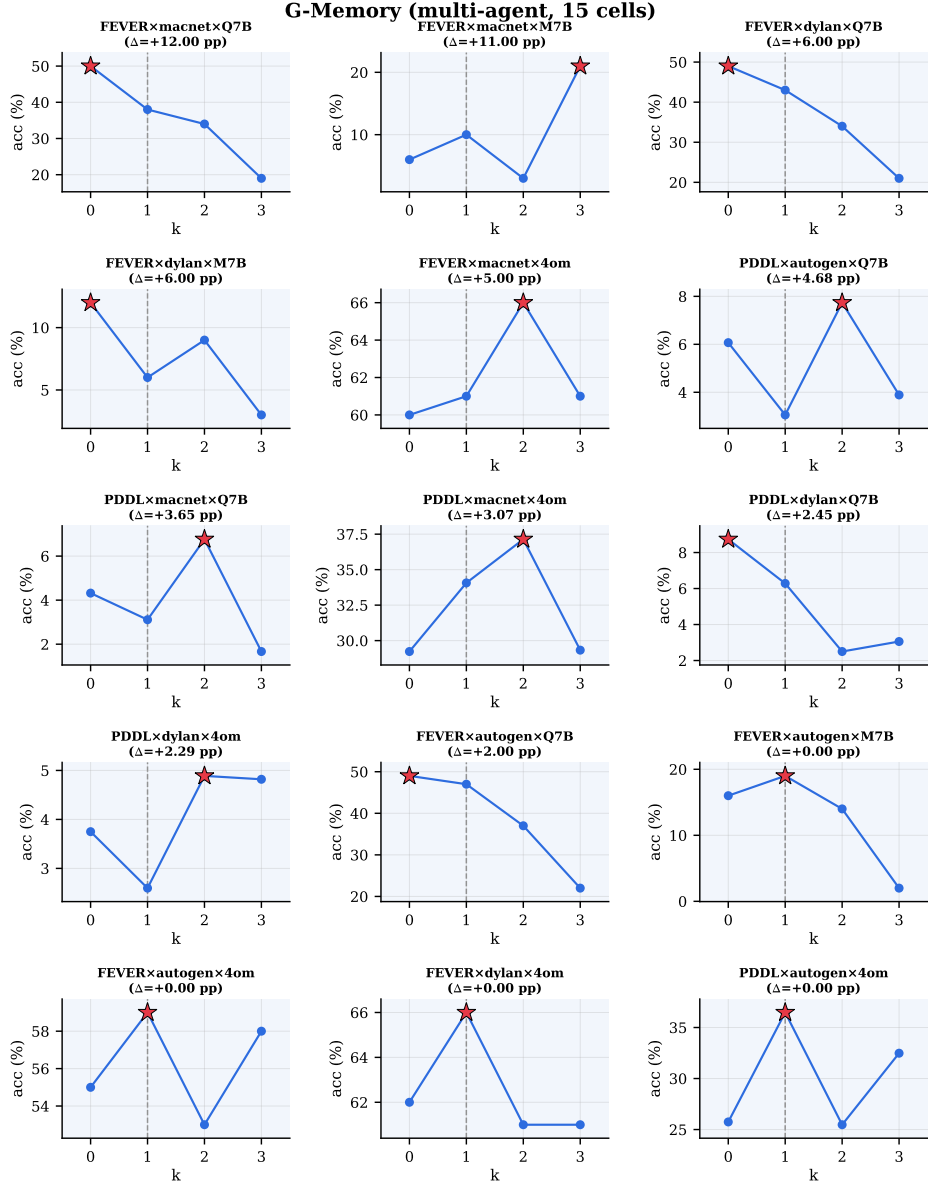


**Figure 4:** Per-cell accuracy vs.  $k$  for all 19 cells. *Top:* 15 G-Memory cells (linear axis, pool  $\subseteq \{0, 1, 2, 3\}$ ). *Bottom:* 4 A-Mem cells (log axis, pool  $\subseteq \{1, 3, \dots, 75\}$ ). Cells ordered by  $\Delta_{vs\ def}$  descending. Dashed line = host default  $k$ ; red star = per-cell oracle  $k$ .

## D.2 Eight Training-Free Bandit Estimators

We compare eight zero-training estimators under the `reliable` bootstrap protocol (Agarwal et al., 2021) on the 19 deployment cells: S0 uniform mean, S1 paired Borda, S2 stratified probe, S3 successive halving, S4 paired LUCB, S5 Empirical-Bernstein race, S6 Hoeffding race, and S7 Thompson (Beta). Tab. 9 reports the inter-quartile-mean (IQM) regret with 95% stratified-bootstrap CIs at the deployment budget  $N=15$ . S0 attains the minimum IQM regret ( $1 \times 10^{-5}$ ), tied by S2/S5/S6/S7 and within CI of S1; only the budget-reallocating S3/S4 are marginally worse, with fully overlapping intervals. This is the quantitative form of the “simpler-is-sufficient” overlap (Fig. 3, main text): no estimator separates from uniform-mean argmax.

**Table 9:** Eight training-free bandit estimators at the deployment budget  $N=15$  (19 cells), `reliable` IQM regret with 95% bootstrap CI. S0 (ours) is tied for the minimum; all intervals overlap.

| estimator              | IQM regret     | 95% CI          |
|------------------------|----------------|-----------------|
| S0 uniform mean (ours) | <b>0.00001</b> | [0.000, 0.0026] |
| S1 paired Borda        | 0.00002        | [0.000, 0.0026] |
| S2 stratified probe    | <b>0.00001</b> | [0.000, 0.0024] |
| S3 successive halving  | 0.00050        | [0.000, 0.0039] |
| S4 paired LUCB         | 0.00097        | [0.000, 0.0039] |
| S5 EB race             | <b>0.00001</b> | [0.000, 0.0026] |
| S6 Hoeffding race      | <b>0.00001</b> | [0.000, 0.0024] |
| S7 Thompson (Beta)     | <b>0.00001</b> | [0.000, 0.0025] |

A paired bootstrap ( $n=2000$  resamples) over the 17 ablation cells corroborates Tab. 9: the sign test on per-cell regret returns  $p_{\text{sign}}=1.0$  against the Empirical-Bernstein race (S0 and S5 commit the identical arm on every cell, so the paired regret difference is exactly zero), and S0’s per-cell wrong-arm rate falls monotonically with  $N$  (e.g. 21.4%  $\rightarrow$  15.2%  $\rightarrow$  8.1% at  $N=5, 15, 30$  on the FEVER/AutoGen/Qwen2.5-7B cell), tracking the  $1/\sqrt{N}$  contraction of  $\rho(N)$ .

## D.3 Sensitivity to Probe Budget $N$

Fig. 5 reports mean stop- $N$  vs  $\Delta_{\text{vs def}}$  for the 19 deployment cells under the  $W_{\text{stable}}=15$  convergence rule, with the probe budget capped at the submitted conservative safety wall  $N_{\text{max}}=200$  (the revised deployment rule derives the wall from cost or target-resolution constraints, Sec. 5.2). G-Memory cells gain monotonically over  $N \in [5, 30]$  and saturate near  $N=30$ ; A-Mem cells saturate at  $N \geq 5$ , so the convergence rule reaches saturation at mean stop- $N \approx 20$ , well within the A-Mem effective cap of

49.

**Probe budget vs improvement (convergence-rule stop)**

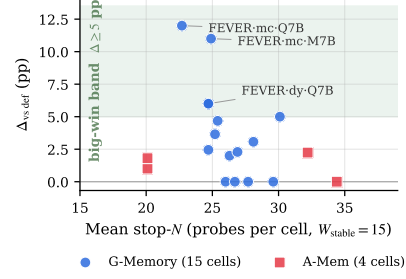


**Figure 5:** Sensitivity to per-cell probe budget: mean stop- $N$  vs  $\Delta_{\text{vs def}}$  for the 19 deployment cells under the  $W_{\text{stable}}=15$  convergence rule.

## D.4 Effect of Candidate-Pool Size $K$

Pool size enters only through the confidence radius  $\rho(N) = \sqrt{\log(2K/\delta)/(2N)}$ , which grows as  $\sqrt{\log K}$ , so the equivalence regime  $\rho(N) > \Delta$  widens with  $K$ , and uniform-mean argmax remains the admissible estimator as the pool grows (Prop. 1). The practical cost is the probe count, which grows linearly in  $K$  because every probe evaluates all candidates jointly. Our pools ( $K \leq 4$  on G-Memory,  $K \leq 11$  on A-Mem) sit well inside the regime; enlarging them changes  $\rho(N)$  only logarithmically while linearly raising probe cost, so the documented host sweeps are the cost-efficient choice.

## E Equivalence Regime Validation

### E.1 Confidence Radius $\rho(N)$ vs. Inter-Arm Gap $\Delta$

The geometric picture of the equivalence regime (Prop. 1) is given in Fig. 6: the Hoeffding confidence radius  $\rho(N) = \sqrt{\log(2K/\delta)/(2N)}$  envelops the observed inter-arm gap  $\Delta$  at every  $N$  used in the deployment regime, so no confidence-bound estimator has signal to eliminate arms.

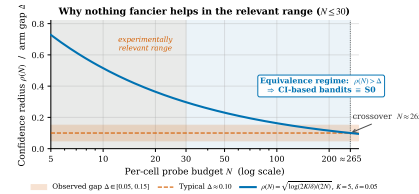


**Figure 6:** Equivalence regime  $\rho(N) > \Delta$  on every cell at the deployment budget.

### E.2 Synthetic $\Delta$ Sweep

Fig. 7 extends the M1 result to a synthetic  $\Delta$  sweep ( $K=5$  paired binary arms,  $\Delta \in [0.01, 0.50]$ ) showing S0 dominance is not an artefact of the 17 abla-

tion cells: S0 sits at or near the minimum wrong-arm rate across the full  $\Delta$  range.

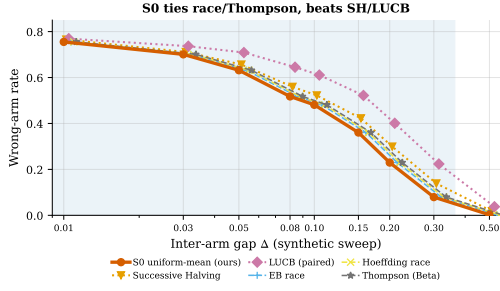


**Figure 7:** M4: synthetic  $\Delta$  sweep ( $K=5$ ,  $N=15$ ). S0 ties race/Thompson estimators and beats SH/LUCB across  $\Delta \in [0.01, 0.50]$ ; the shaded band marks the equivalence regime  $\Delta < \rho(N)$ .

### E.3 Budget Waste and Probability-of-Improvement (M2/M3)

Two supporting layers of evidence for Prop. 1 beyond the M1 wrong-arm overlap. (a) Adaptive bandits leave probe budget unspent: Successive Halving consumes 7/15 and LUCB 11.6/15, while S0 (and all confidence-bound estimators inside the regime) uses the full budget. (b) Per the reliable bootstrap protocol (Agarwal et al., 2021), the probability that S0 has wrong-arm rate at or below each baseline ties or exceeds 0.50 for every alternative estimator; S0 is never strictly dominated.

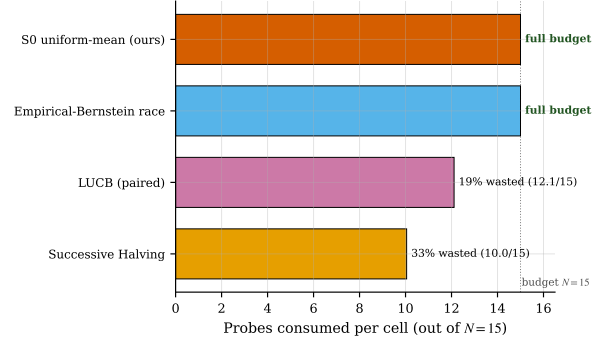
## F Cross-Cell Shortcut Baselines

A natural objection to per-cell warm-up is that a cheaper shortcut might predict  $k^*(c)$  from observable cell features or from other cells, avoiding the per-cell probe. This appendix shows both shortcuts fail.

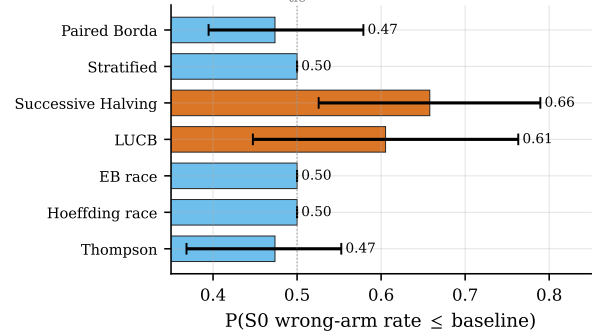
### F.1 Leave-One-Cell-Out Prediction

Tab. 3 (main text) reports the leave-one-cell-out (LOOCV) comparison: a model trained to predict  $k^*$  from every other cell and applied to the held-out cell. No cross-cell predictor matches the per-cell oracle, confirming that the  $k^*$  signal does not transfer across cells, consistent with the structural model of App. A.1, where unobserved factors  $(\eta, \kappa, \delta)$  drive  $k^*$ .

**Reporting protocol for Tab. 3.** Every method commits a single  $k$  per cell, and the table reports the mean per-cell % of oracle of that committed  $k$ , together with its  $p_{25}$ ,  $p_{10}$ , and the number of cells clearing the  $\geq 95\%$  success bar, over the 15 G-Memory cells; Adaptive- $k$  is scored only on the



(a) M2: adaptive bandits underspend the per-cell budget.



(b) M3:  $P(S0 \text{ wrong-arm rate} \leq \text{baseline})$  across bootstrap iters.

**Figure 8:** M2+M3 evidence. (a) Adaptive bandits leave probe budget unspent. (b) Bootstrap PoI shows S0 ties or beats every alternative estimator.

3 A-Mem LoCoMo cells whose per-item similarity scores it consumes. The lookup tables and the logistic-regression gate obtain their  $k$  for a held-out cell by LOOCV prediction from the remaining cells, whereas warm-up obtains its  $k$  from the held-out cell’s own deployment-time probe pass (Sec. 5.3). The warm-up rows are scored per-cell under single-evaluation scoring (no multi-seed averaging), which is why they sit slightly below 100%; under the matched multi-seed protocol that defines the oracle, warm-up returns the committed arm exactly (100%, Tab. 2).

### F.2 Observable Features Fail to Predict $k^*$

Regressing  $k^*$  on {model size, task, coordinator, retrieval cosine mean / std, log store size} gives a joint OLS  $R^2$  of 0.71 in-sample but collapses to 0.18 under LOOCV (Fig. 9); no individual feature exceeds  $R^2 = 0.21$ . The training-free score-geometry port (App. G, F1) attains only 60.0% of the per-cell oracle on the A-Mem cells with well-defined oracle, with score-gap-to-oracle Spearman  $\rho \approx 0.08$ .

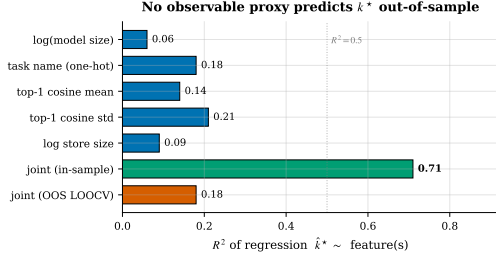


**Figure 9:** Observable proxies fail to predict  $k^*$ : joint  $R^2$  drops from 0.71 in-sample to 0.18 under LOOCV.

## G Adaptive Retrieval Methods Analysis

Sec. 2.2 groups single-turn adaptive-retrieval techniques into five families and lists two deployment constraints (signal availability and deployment cost) that prevent direct transfer to agent memory. This appendix expands each family with the per-method details that the main-text discussion compresses; the consolidated differentiation table (Tab. 1) is in the main text.

**F1. Training-free per-query  $k$  from score geometry.** Adaptive- $k$  (Taguchi et al., 2025) cuts at the steepest drop in the sorted similarity scores; CAR (Xu et al., 2025b) detects the cluster transition in the score-distance distribution. Adjacent to F1, TARG (Wang et al., 2026) proposes a training-free per-query gate that decides *whether* to retrieve rather than how many items to retrieve. Both Adaptive- $k$  and CAR consume per-item similarity scores that graph-structured multi-agent memory does not expose, rendering them inapplicable to G-Memory cells; on A-Mem cells the signal is available but, as Tab. 3 (last row) reports, the empirical port attains only 60.0% of the per-cell oracle on the 3 A-Mem LoCoMo cells with well-defined oracle, with Spearman correlation between the score-gap signal and the per-query oracle- $k$  averaging  $\rho \approx 0.08$  ( $|\rho| \leq 0.11$  on every cell, all  $p > 0.4$ ).

**F2. Learned query routing.** Adaptive-RAG (Jeong et al., 2024) trains a T5-large complexity classifier that routes each query among non-retrieval, single-step, and multi-step strategies. The router requires labelled training data that agent-memory benchmarks do not ship, and its per-query routing inflates LLM invocations linearly with the agent’s interaction turns.

**F3. Self-reflective iterative retrieval.** Self-RAG (Asai et al., 2024) fine-tunes the LLM to emit reflection tokens that trigger re-retrieval; FLARE (Jiang et al., 2023b) re-retrieves when

next-token confidence falls below a threshold; SeaKR (Yao et al., 2025) consults internal-state confidence to decide whether to retrieve further. All three multiply LLM-invocation cost by the agent’s turn count and rely on per-query (or per-token) state that is incompatible with committing a single  $k$  for the deployment lifetime.

**F4. Learned ranking and candidate-set selection.** RankRAG (Yu et al., 2024) jointly trains an instruction-tuned ranker and generator; DynamicRAG (Sun et al., 2025) performs LLM-feedback-driven adaptive reranking; DPS (Meng et al., 2025) and OptiSet (Jiang et al., 2026) learn end-to-end set selectors. All require labelled training data and operate at per-query granularity.

**F5. Static grid search / lookup on a held-out validation set.** The implicit baseline behind fixed- $k$  deployment. Fig. 1 and Tab. 2 refute it directly: across the heterogeneous deployment grid, no single  $k$  matches the per-cell best on more than a fraction of cells, and weak-LLM cells suffer a within-pool collapse on the upper end of  $\mathcal{K}_c$ . The theoretical and statistical grounds are given in App. A and App. F.2.

**Why adaptive- $k$  does not transfer to agent memory.** The two constraints are structural, not incidental. F1’s score geometry is unavailable on graph-structured memory and uninformative on the memory it is available for ( $\rho \approx 0.08$ ); F2–F4 require labelled training data and per-query LLM calls that scale with the agent’s lifetime; F3’s per-token signal cannot be reduced to a single committed  $k$ . Per-cell warm-up sidesteps all of these: it needs no scores, no labels, no auxiliary model, and commits once per cell before any rollout.

**Differentiation summary.** The consolidated comparison across the four transferability axes is reported in Tab. 1 (main text, Sec. 2.2).