SIGNAL BEFORE SCALE: USER HISTORY AND INFERENCE COMPUTE IN LLM PERSONALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalizing a language model consumes two resources with unrelated cost structures: information about the user, and compute at inference time. Their marginal returns have been measured separately; their interaction has not, leaving open whether compute can buy back missing user signal. We cross a per-user signal axis against an inference-compute axis. The signal axis is the number $k \in \{0, \ldots, 32\}$ of the user's own profile items retrieved into the prompt, manipulated within user rather than observed across users, removing the confound with activity level. The compute axis is *selector-free*: three model tiers and a chain-of-thought sampling budget over a frozen model. The design spans four LaMP tasks and 11,197 calls. On the three tasks where retrieved history moves the metric, eight items capture at least 96% of the total gain. At matched sample size, 39 of 71 signal contrasts have 95% bootstrap intervals excluding zero, against 10 of 82 for compute. On the citation-choice and rating-prediction tasks, the smallest model given eight of the user's own items outperforms the largest given none; on the latter it halves the error. The interaction is resolvable, and its sign depends on the task: model scale and user history substitute on rating prediction and complement on movie tagging, at four items and above. Any account of when inference compute pays for personalization must be task-conditional. A direct measurement explains why: at temperature 1.0, 38–67% of items return an identical answer on every sample, leaving self-consistency nothing to aggregate until history is in context. With no trained selector present, the shortfall lies in the candidate pool rather than in selection. Signal before scale: extra compute needs signal before it pays.

1 INTRODUCTION

A deployed personalization system spends two kinds of resource. It conditions on information about the user — past reviews, past choices, past writing — and it spends compute at inference time to turn that information into an answer. The two are constrained differently: retrieved history is bounded by what the user has actually done and by what a privacy budget permits, while inference compute is bought per token from a menu of model tiers and sampling budgets. An operator choosing where to put the next unit of effort needs to know which axis binds.

Both axes have been measured, separately. On the signal side, LaMP (Salemi et al., 2024b) established the protocol we use, and Salemi & Zamani (2025) already report the *direction* of our signal-side finding — though at a fixed retrieval budget and with signal volume as an *observed user attribute* rather than an intervention. On the compute side, quality is known to rise with sampling budget on tasks with verifiable answers (Wang et al., 2022; Brown et al., 2024; Snell et al., 2024). Most directly, Zhang & He (2026) scale inference compute *for personalization* and bound even oracle selection at $O(\sqrt{\ln N})$ in the number of samples. That ceiling is prior work establishing the compute marginal, and it is evidence for the thesis of this paper: that axis has a known, slow-growing ceiling. We do not claim to introduce a test-time scaling law for personalization; that claim belongs to Zhang & He (2026). Full comparisons are in §2.

What has not been measured is the **cross-term**. Does compute buy back missing user signal? Does the marginal value of a bigger model rise or fall as a user's history is filled in? Both are questions about a two-dimensional surface, answered only by running the grid rather than either marginal.

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

What we run. The signal axis is $k \in \{0, 1, 2, 4, 8, 16, 32\}$, the number of the user's own profile items retrieved by BM25 (Robertson & Zaragoza, 2009) into the prompt; $k{=}0$ is the no-signal condition, and k is manipulated within user, so the axis is not confounded with whatever else distinguishes prolific users from sparse ones. The compute axis is three model tiers of one frozen model family plus a 3–4 sample chain-of-thought budget with self-consistency. It is deliberately *selector-free*: no trained personalized reward model anywhere in the pipeline. This is the regime most deployments sit in, and the one in which the reward-model pathologies Zhang & He (2026) diagnose (user-level reward collapse, query-level reward hacking) cannot explain compute insensitivity, because there is no reward model to collapse. The grid is 11,197 usable model calls on four LaMP tasks.

Contributions, each at the strength the measurements support and no stronger. **(1) The surface, and early saturation on the signal axis** (§4): the knee — the smallest k reaching 90% of the total $k{=}0 \to 32$ gain — sits at $k{=}8$ on LaMP-1/2/3 though only LaMP-1's and LaMP-3's intervals localise it, at least 96% of the gain is realised by $k{=}8$ on the three tasks where that gain resolves, and on LaMP-3 one irreducible-error floor shared by all three tiers improves held-out prediction over per-tier floors ($t{=}-4.41$), so a larger tier buys a head start, not a lower floor. **(2) The signal axis dominates the compute axis, and the comparison behind that claim is two effects in one** (§4): at equal n, signal contrasts resolve 39/71 against 10/82 for compute. A small model with history beating a large model without it sums retrieval's gain with the large model's pre-existing lead, so it shows substitution only where that lead is real: on LaMP-3 a 0.172 MAE deficit is overturned by a 0.313 gain. Added signal then *shrinks* the larger tier's advantage there and *grows* it on LaMP-2 — the interaction differs in direction, not size. LaMP-5, where retrieval does not move the metric, is the scope limit. **(3) A selector-free mechanism, and a decision rule that saves compute rather than quality** (§5, §7): at temperature 1.0, 38–67% of items return an identical answer on every sample, so self-consistency's gain resolves only at $k{=}32$; and on rating prediction a threshold on *observable* signal volume cuts upgrades from 100% to 71% while beating a matched-budget random policy — though it does not beat always upgrading, and does not transfer.

The slogan that survives all four tasks is not "compute does not help" — on LaMP-2 it helps more once signal is present, and on LaMP-5 signal helps not at all. It is *signal before scale*: wherever either axis moves the metric, the signal axis moves it first, further and more reliably, and the return to extra compute is contingent on signal already being there.

2 Related work

Personalizing LLMs from user history. LaMP (Salemi et al., 2024b) established the protocol we use: each query is paired with a user profile and a retriever selects profile items to condition the generation. Later work optimizes retrieval (Salemi et al., 2024a), extends to long-form generation (Kumar et al., 2024), replaces retrieval with per-user adapters (Tan et al., 2024), or compresses history into soft prompts (Liu et al., 2025a); Zhang et al. (2024) surveys the space and Zhang et al. (2025) the adjacent agent-memory literature. A recurring theme is that users differ enormously in how much history they provide. Salemi & Zamani (2025) make this explicit, finding that parameter-efficient tuning benefits more as available user data grows while retrieval is the stronger cold-start choice; Zhang et al. (2026) build a method around most users supplying only a handful of interactions, Qu et al. (2025) address cold start through online preference feedback, and Sun et al. (2026); Sawant (2026) document failure modes. These results treat signal volume as an observed user attribute at a fixed retrieval budget, and Salemi & Zamani (2025) in particular already reports the *direction* of our signal-side result, which we therefore do not present as new. What we add is control and a second axis: we vary k within user, so the signal placed in context is manipulated rather than correlated with activity level, topic diversity or account age, and we cross it with inference compute.

Spending compute at inference time. Self-consistency (Wang et al., 2022), repeated sampling (Brown et al., 2024), compute-optimal test-time allocation (Snell et al., 2024), budget-controlled reasoning (Muennighoff et al., 2025; Ma et al., 2025) and inference-time scaling for reward models (Liu et al., 2025b) show that on tasks with verifiable answers quality improves with sampling or reasoning budget, sometimes more efficiently than parameter scaling. The dividend is domain-dependent: Kour et al. (2025) report that added reasoning and self-reflection give only limited gains

Table 1: The four grids. **A** is the primary surface, on which every headline number is computed; **B** varies the sample budget to expose the mechanism of §5; **C** and **R** are the two controls of §6.

Grid	Design	Prompting	Cells	n/cell
A primary surface	$7\,k \times 3$ tiers $\times$ 4 tasks	direct, one greedy sample	83 of 84	90–99
B mechanism	$3\,k \times 1$ tier $\times$ 2 tasks	CoT, T=1.0, **3–4 samples**	6	39
C CoT control	$3\,k \times 1$ tier $\times$ 4 tasks	CoT vs. direct, **one greedy sample**	11 of 12	60–99
R retriever control	$6\,k \times 1$ tier $\times$ 2 tasks	recency vs. BM25, matched k	12	99

on subjective preference, opinion and belief questions — prior evidence that the return shrinks where the target is a subjective property rather than a verifiable answer.

In personalization specifically, Zhang & He (2026) is the closest prior work and establishes the compute marginal: sampling from a personalized policy with personalized-reward-model selection, an oracle ceiling of $O(\sqrt{\ln N})$ (their Thm. 3.1, under a sub-Gaussian assumption), a scaling law that multiplies that ceiling by the reward model's correlation with true quality, and user-level reward collapse and query-level reward hacking as the failure modes that keep practical reward models below it. Our study is complementary: their compute axis is *reward-model-mediated* and their diagnostic concerns a trained selector failing to realise an oracle ceiling, whereas ours is selector-free — model tier and sample budget over a frozen model — so their failure modes cannot explain insensitivity in our regime, while their $O(\sqrt{\ln N})$ ceiling is a reason to expect it. Positive test-time-scaling results also exist on the recommendation side, via PRM-guided beam search (Guo et al., 2026) and via prediction merging (Lyu et al., 2025); those concern semantic-ID generation and CTR-style prediction rather than LLM text personalization, and we scope our claims to the latter. Inference-time steering targets preference control without retraining (Su et al., 2026; Luo et al., 2026; Han et al., 2024), and Liu et al. (2025a) already uses the phrase "scaling laws" in a personalization-prompting context — we make no priority claim on that pairing.

Scaling laws and their limits. Power-law scaling with an irreducible-error term (Kaplan et al., 2020; Hoffmann et al., 2022) and its theoretical accounts (Bahri et al., 2024) predict saturation once a resource stops binding, and real learning curves break rather than follow one exponent (Caballero et al., 2022) — which is why we report a knee, not an exponent (§4.2). Most directly, Muennighoff et al. (2023) show that when data rather than compute is scarce, compute over a fixed data pool yields decaying returns; ours is the inference-time, per-user analogue, with the user's own history as the scarce pool. Analogous scaling studies exist for industrial recommenders consuming user history (Hou et al., 2026; Lai et al., 2025), but concern training compute for specialized architectures. We make no information-theoretic claim: a targeted search returned no prior result formalizing a per-user information ceiling for LLM personalization, and we establish none.

3 Experimental design

Tasks and metrics. Four LaMP (Salemi et al., 2024b) dev splits spanning classification, ordinal regression and free-text generation: LaMP-1 citation choice (accuracy over two candidates), LaMP-2 movie tagging (macro-F1 over 15 tags), LaMP-3 product rating (MAE on the 1–5 scale) and LaMP-5 scholarly title generation (ROUGE-1 F). Two conventions affect comparability. First, "LaMP-2" names two distinct tasks in this literature; we use the movie-tagging variant, and because we could not establish which variant a given published number refers to, we table no result against a published baseline (Reproducibility Statement). Second, our LaMP-2 macro-F1 is computed over the closed 15-tag set rather than the union of gold and predicted labels, so the denominator is fixed rather than prediction-dependent (App. B).

The signal axis. For each query we retrieve the top-k items of that user's own profile with Okapi BM25 (Robertson & Zaragoza, 2009), $k \in \{0, 1, 2, 4, 8, 16, 32\}$, and insert them into the prompt; k=0 is the no-signal condition. Because k varies *within* user, the signal axis is separated from user activity level. We also record $\mathrm{eff}_k = \min(k, |\text{profile}|)$, the items that actually reach the context. Only LaMP-2 binds: at k=32, 60% of its users have shorter profiles than the budget and mean eff_k is 22.1, whereas LaMP-1/3/5 never cap (App. I). This matters twice below — LaMP-2's flat high-k behaviour is partly a ceiling on *available* signal, and the rule of §7 is keyed on eff_k rather than k.

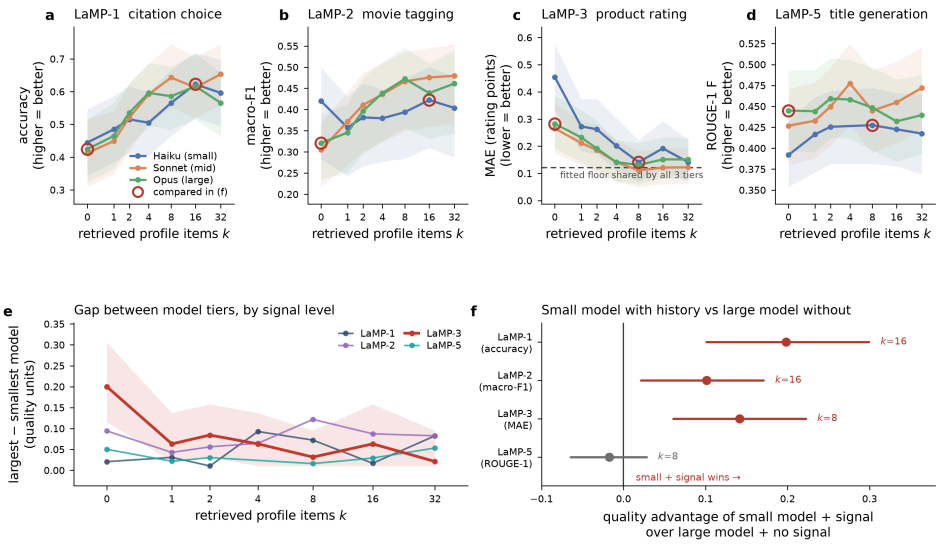

Figure 1: The grid-A surface. **(a–d)** Headline metric against retrieved profile items k, one line per model tier, shaded 95% bootstrap intervals, n=93–99 items per cell; the dashed line in (c) is the single irreducible-error floor shared by all three tiers, fitted in §4.2. **(e)** Largest minus smallest tier at fixed k, in quality units, paired over the items shared by all 21 cells of a task (n=77–98). **(f)** The smallest model at its best k against the largest model at k=0, paired, n=99; red marks intervals excluding zero. The circled points in (a–d) are the two cells compared in (f).

The compute axis, and what it is not. Two components: three tiers of one frozen model family, Haiku (small), Sonnet (mid) and Opus (large) — a ladder with two model-scale steps — and 3–4 chain-of-thought samples at temperature 1.0 with majority-vote self-consistency (Wang et al., 2022). Both are *selector-free*: no trained personalized reward model, no Best-of-N selection, no learned verifier. The axis does *not* include native reasoning-token or thinking budgets — the serving interface exposes none and both configured GPU hosts were unreachable, so all runs used hosted APIs. Nothing here speaks to reasoning-token budgets or to reward-model-mediated Best-of-N.

Subsets, grids and inference. Each task contributes a 150-query seeded subset stratified into equal-sized profile-length terciles, after removing gold-leakage items (5 LaMP-1, 9 LaMP-5) — retrievable for free at high k, and so able to manufacture exactly the k-dependent gain we measure — and one malformed input. A **cell** is one experimental condition — a (task, tier, k, prompting variant, retriever, sample budget) — evaluated on every item of that grid's subset; the sample budget is part of its identity, and separates grids B and C. Each cell of grids A, C and R runs 99 items, 33 per stratum. Table 1 gives the four grids: grid A is the primary surface, grid B varies the sample budget to expose the mechanism of §5, and grids C and R are the two controls of §6. Together they are 11,394 responses, 11,197 usable, 14 unparseable (0.13%); parse failures are reported per cell and excluded from metric averages rather than scored as wrong, so no result rests on a parser artefact. All contrasts are paired percentile bootstraps over items (2000 resamples, seed 12345) restricted to query ids present in both cells; a contrast is *resolvable* when its 95% interval excludes zero, and we report n with every headline number. Functional forms are compared by paired repeated 5-fold cross-validation over items with identical folds. We do not correct for multiplicity: the resolvable-effect counts below compare two families of identically computed contrasts. Full settings, exclusions and losses: Reproducibility Statement.

4 THE SIGNAL–COMPUTE SURFACE

Figure 1(a–d) is the object this paper was built to produce. Read down a column for the effect of buying a larger model at fixed user signal; read along a row, the effect of retrieving more of the user's own history at fixed compute.

Table 2: Saturation of the signal axis. Total gain is the pooled $k{=}0 \to 32$ change in the headline metric, in quality units (larger is better on every task), pooled over the three tiers and bootstrapped over the items shared by all 21 cells of that task. The knee is the smallest k at which the running maximum of the quality curve reaches 90% of the total gain. Brackets are 95% percentile bootstrap intervals over items.

Task	Metric	n	Total gain $k{=}0 \to 32$	Knee k (90%)	Fraction of gain by $k{=}8$
LaMP-1	accuracy	97	0.1821 [0.1100, 0.2543]	8 [4, 16]	0.9623 [0.7586, 1.2381]
LaMP-2	macro-F1	77	0.1128 [0.0335, 0.1730]	8 [1, 32]	1.0028 [0.6931, 1.8808]
LaMP-3	MAE	95	0.2105 [0.1404, 0.2842]	8 [4, 8]	1.0667 [0.9104, 1.2546]
LaMP-5	ROUGE-1	98	0.0207 [-0.0096, 0.0528]	2 [1, 32]	0.8329 [-1.6178, 4.0631]

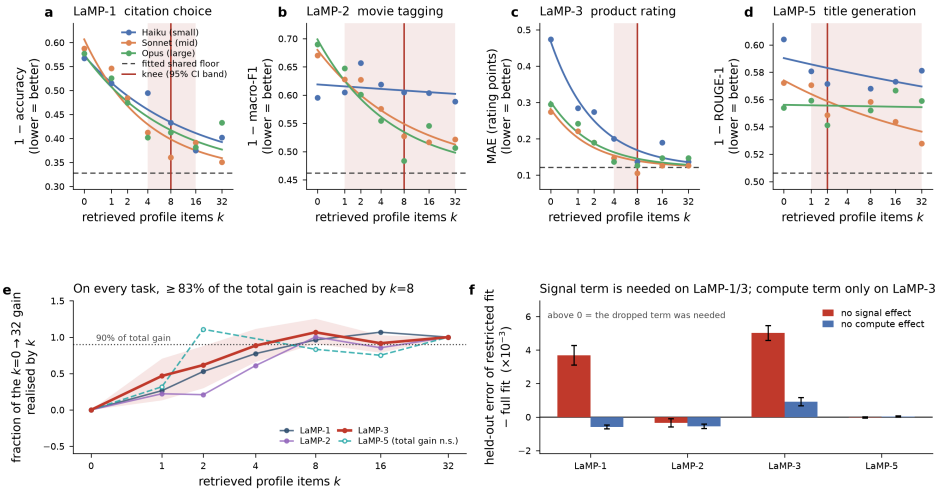

Figure 2: Saturation fits. **(a–d)** Observed cell error (points) and the shared-floor fit $\mathrm{err}(k) = e_\infty + a_m(k{+}1)^{-b_m}$ (lines); the vertical line and band are the knee and its 95% interval, the dashed line the fitted shared floor. **(e)** Fraction of the total $k{=}0 \to 32$ gain realised by k; the dotted line is the 90% knee criterion. The "$\geq$ 83%" in that panel's own title is the minimum *including* LaMP-5, whose fraction is a ratio with a denominator that crosses zero and is therefore undefined rather than small; across the three tasks where the total gain resolves the minimum is 96%. **(f)** Increase in held-out error when a term is dropped from the full fit; above zero means the dropped term was needed. Error bars are paired standard errors over 50 cross-validation folds.

4.1 The signal axis saturates early

On LaMP-1, LaMP-2 and LaMP-3 the total gain from retrieving user history is resolvable and the knee sits at $k{=}8$ (Table 2); the LaMP-3 interval is tight ($[4, 8]$), LaMP-1's is $[4, 16]$ and LaMP-2's is uninformative ($[1, 32]$). On those **three** tasks at least 96% of the total gain is already realised by $k{=}8$ (Fig. 2e). On LaMP-5 the total gain is **not** resolvable (0.021 $[-0.010, 0.053]$ ROUGE-1, $n{=}98$), so its knee is meaningless and its fraction of gain is *undefined* — the denominator of that ratio vanishes in 8.8% of resamples, giving the interval $[-1.62, 4.06]$ — and averaging it in is what would lower the figure above to 83%. We report it as such: free-text scholarly title generation is a task on which retrieving the author's own past titles does not move the metric within our resolution. We keep LaMP-5 throughout as the scope limit rather than dropping it. Every claim below that depends on the signal axis moving is a claim about the three tasks where it demonstrably does.

4.2 The fit identifies a shared floor, not a scaling exponent

We fitted a full per-tier saturating model to the 21 cell errors of each task (7 values of $k \times 3$ tiers) and compared it against six restrictions of itself by paired repeated cross-validation over items, with identical folds (definitions in Table 8, results in Table 9).

The signal term is needed; the compute term usually is not. Dropping the signal effect costs held-out accuracy on LaMP-1 ($+3.69 \times 10^{-3}$ MSE, $t{=}{+}6.32$) and LaMP-3 ($+5.01 \times 10^{-3}$, $t{=}{+}11.26$).

Dropping the compute effect costs nothing on LaMP-1 (t=−5.13), LaMP-2 (t=−4.18) or LaMP-5 (t=+0.70) — on those tasks a fit with no model-tier term predicts held-out items as well or better — and is needed only on LaMP-3 ($+9.16 \times 10^{-4}$, t=+3.81); see Fig. 2(f).

A bigger model does not buy a lower floor. Forcing one irreducible-error term e_∞ across all three tiers *improves* held-out prediction on LaMP-3 (MSE lower by 1.54×10^{-4}, paired SE 3.5×10^{-5}, t=−4.41 over 50 folds) and LaMP-2 (t=−4.48), and is neutral on LaMP-1 (t=−1.22) and LaMP-5 (t=+1.89): on no task is a tier's own floor worth its two extra parameters. On LaMP-3, where the fit is identified, that floor is $e_\infty = 0.1206$ $[0.0323, 0.1808]$ MAE and the tiers differ mainly in amplitude ($a = 0.348/0.162/0.181$ for small/mid/large, not monotone in tier; Table 10); their exponents the data cannot separate ($b \approx 0.90$–0.98, intervals spanning $[0.28, 2.26]$). What a tier buys is a head start, and on LaMP-3 the head start is spent at once: the tier spread falls from 0.200 $[0.116, 0.305]$ MAE at k=0 to 0.063 at k=1 and 0.021 $[0.011, 0.095]$ at k=32 (Fig. 1e), three quarters of the collapse coming from the first retrieved item. That collapse is a LaMP-3 result: the spread does not shrink on LaMP-2 ($0.094 \rightarrow 0.082$ macro-F1) or LaMP-5 ($0.050 \rightarrow 0.053$ ROUGE-1), and on LaMP-1 it never exceeds 0.093 accuracy at any k. Across the grid the tier spread exceeds 0.13 quality units exactly once, on LaMP-3 at k=0.

We cannot claim a clean power law, and do not. The per-tier free-floor fits are under-identified on three of four tasks — e_∞ intervals reach $[0.00, 0.72]$ and exponent intervals hit the optimiser bounds — and an exponential form $e_\infty + a\exp(-k/\tau)$ is indistinguishable from the power form on LaMP-3 (t=+0.06) and marginally preferred on LaMP-1 (t=−2.03), the behaviour Caballero et al. (2022) document for real learning curves. Our data resolve *that* the signal curve saturates and roughly *where*, not its algebraic form; we therefore quote the knee, plus the LaMP-3 shared floor, and no exponent. That floor is a floor at $k \leq 32$ with BM25 retrieval and these three tiers — not an information-theoretic bound, of which we establish none.

4.3 Signal contrasts resolve at equal n; compute contrasts mostly do not

Two comparisons, neither needing a functional form (both tabulated in App. G). First, a head-to-head. **The smallest model with retrieved history beats the largest model with none on three of four tasks.** On LaMP-3, Haiku at k=8 reaches MAE 0.1414 $[0.0707, 0.2222]$ against Opus at k=0 at 0.2828 $[0.1919, 0.3838]$; the paired difference is -0.1414 MAE $[-0.2222, -0.0606]$, n=99, p=0.0005 — the small model given eight of the user's own reviews halves the error of the largest model given none. On LaMP-1 the advantage is $+0.198$ accuracy $[0.101, 0.299]$ at k=16 (n=99, p=0.0005), on LaMP-2 $+0.102$ macro-F1 $[0.021, 0.171]$ at k=16 (n=99, p=0.006), and on LaMP-5 -0.017 ROUGE-1 $[-0.065, 0.028]$ (n=99, p=0.465): not resolvable, and reported as the negative case, which scopes the claim to tasks where retrieved history moves the metric at all. But the head-to-head isolates *substitution* only where the large tier actually leads at k=0 (Table 12): it does on LaMP-3 (+0.172) and LaMP-5 (+0.053), while on LaMP-1 (−0.020) and LaMP-2 (−0.099) the small tier already leads, so those two report a signal gain and a tier ordering — on LaMP-2 the small tier's own gain is -0.001 $[-0.079, 0.103]$, for the reason §4.4 gives.

Second, a count, which requires no choice of k: every signal contrast (k vs. k=0 at fixed tier) against every compute contrast (tier vs. tier at fixed k), on the same items at $n \approx 99$ under identical bootstrap settings. **Signal contrasts resolve 39 times in 71 attempts (55%), compute contrasts 10 times in 82 (12%)** — four and a half times as often, under the same test — and on LaMP-1 not one of the 21 tier-vs-tier contrasts resolves at any k. The gap is not an artefact of unequal spans. Our ladder is short, two model-scale steps against the signal axis's $0 \rightarrow 32$, but on LaMP-1 and LaMP-2 *no* position on it reaches the signal span at all (Table 12): the compute axis does not merely fall short of the signal axis, it does not arrive.

The exchange rate, where one exists. Dividing the signal span *at the small tier* by the per-step compute gain at k=0 gives an exchange rate. That span is the small tier's own, not the tier-pooled gain of Table 2: on LaMP-2 the two carry opposite signs (-0.016 against $+0.113$) because the small tier gains nothing from retrieval on that task. **On LaMP-1 and LaMP-2 the compute span at k=0 is itself non-positive** (-0.020 accuracy, -0.099 macro-F1), **so no number of model-scale steps matches the signal gain on either task**. On LaMP-3, where the span is positive, matching the k=0 → 32 signal span of 0.3131 MAE would take 3.65 model-scale steps on a ladder that has two — but the two rungs are not interchangeable: Haiku→Sonnet lowers MAE by 0.189 while

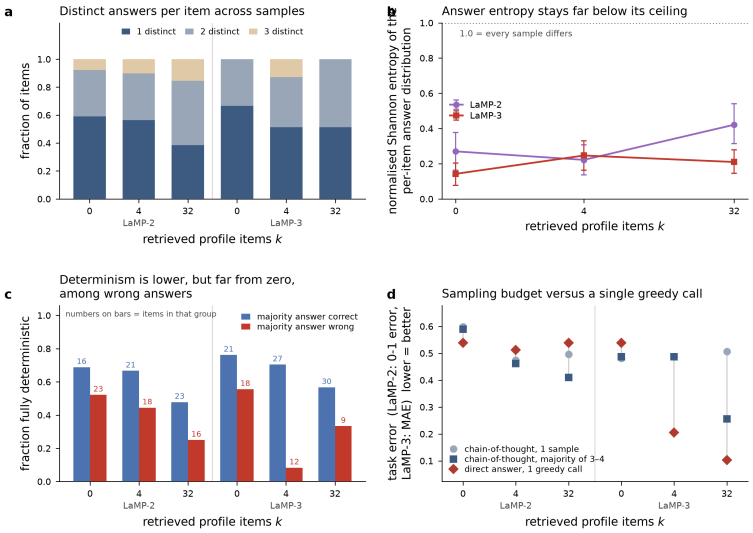

Figure 3: Grid B: chain-of-thought at temperature 1.0, small tier, n=39 items per cell. **(a)** How many distinct parsed answers an item's 3–4 samples produced: *1* means every sample returned the identical answer (*fully deterministic*), and that band is the quantity (c) splits by correctness; *3* is the maximum observed anywhere in the grid. **(b)** Mean normalised Shannon entropy of the per-item answer distribution; 1.0 would mean every sample differed. **(c)** Fraction fully deterministic, split by whether the majority answer was correct; group sizes on the bars. **(d)** Task error for one chain-of-thought sample, the majority of 3–4, and a single greedy direct call on the same items. Per-cell values with intervals: App. F.

Sonnet→Opus *raises* it by 0.018, so the per-step figure averages a rung that helps with one that hurts. We therefore read 3.65 as **no position on this ladder reaches the signal gain** rather than as a multiple to be extrapolated (§8). On LaMP-5 the rate is 0.96, nominally interchangeable, but both spans are under 0.06 ROUGE-1 and neither is resolvable.

4.4 The interaction is resolvable, and its sign is task-dependent

The cross-term is the object we came for, and it does not have one sign. We measure it as a difference-in-differences on the shared query ids, $[M(\text{large}, k) - M(\text{large}, 0)] - [M(\text{small}, k) - M(\text{small}, 0)]$, converted to quality units by the metric's sign (App. G).

On **LaMP-3 the axes substitute**: the advantage of Opus over Haiku falls by 0.182 MAE $[0.061, 0.323]$ between k=0 and k=32 (n=99, p=0.003), and by 0.186 $[0.082, 0.299]$ for Sonnet over Haiku (n=97, p=0.0005), resolvable at every $k \in \{4, 8, 16, 32\}$. On rating prediction the marginal value of model scale collapses once user history is in context. On **LaMP-2 they are complements — the opposite direction**: the Opus-over-Haiku advantage *grows* by 0.157 macro-F1 $[0.022, 0.249]$ (n=99, p=0.016) and by 0.167 $[0.049, 0.252]$ for Sonnet over Haiku (n=93, p=0.002), again resolvable at every k. That is not a macro-F1 artefact: on plain accuracy, Opus minus Haiku moves from -0.010 at k=0 to $+0.121$ at k=32. The larger tiers extract more from 32 retrieved movie tags than the small one does. On **LaMP-1 there is nothing to interact** (DiD -0.010 $[-0.152, 0.111]$, p=0.886; the compute axis has no resolvable effect at any k), and on **LaMP-5 the effect is weak** — the only resolvable DiD is Opus–Haiku at k=16 (-0.044 $[-0.087, -0.002]$ ROUGE-1) and every effect is under 0.05.

Seventeen of 46 DiD contrasts are resolvable. We draw the conclusion the data supports and no more: **the direction of the interaction between user signal and inference compute is task-dependent**. A sentence of the form "compute substitutes for signal" is true on rating prediction, false on movie tagging, and untestable on the other two for want of a compute effect. What generalises is the ordering, not the sign of the cross-term. App. E offers a conjecture about why these two tasks differ; we did not design an experiment to test it.

5 MECHANISM: THE ANSWERS BARELY MOVE

Why would extra compute buy so little? For the sampling component of the axis we can answer directly, without a selector and therefore without any reward-model pathology to appeal to: **there is almost nothing to sample over.** At temperature 1.0 with chain-of-thought prompting, **38% to 67% of items are *fully deterministic*** — an identical parsed answer on every one of their 3–4 samples (LaMP-3 k=0: 0.667 $[0.513, 0.821]$; LaMP-2 k=32: 0.385 $[0.231, 0.538]$) — mean normalised answer entropy runs 0.14 to 0.42 — 58% to 86% below its ceiling — and **no item in any cell ever produced more than three distinct answers** (Fig. 3a,b; Table 11). Self-consistency needs disagreement to aggregate, and at low k it is not there.

The measured gains follow. Majority vote over 3–4 samples against the mean single sample is **not resolvable at k=0 or k=4 on either task**: LaMP-3 -0.006 MAE $[-0.071, 0.058]$ at k=0 and 0.000 $[-0.096, 0.083]$ at k=4; LaMP-2 $+0.009$ accuracy $[-0.051, 0.068]$ and $+0.013$ $[-0.051, 0.083]$. It becomes resolvable only at k=32 (LaMP-3 $+0.250$ MAE $[0.096, 0.423]$; LaMP-2 $+0.086$ accuracy $[0.017, 0.162]$). This is the same asymmetry as §4.3, not a counterexample to it: **extra sampling needs signal before it pays.**

The k=32 sampling gain is recovery, not progress. On the same 39 items at k=32, four-sample chain-of-thought majority vote reaches MAE 0.2564 while *one greedy direct call* reaches 0.1026 (Fig. 3d): four times the budget, 2.5× worse than one deterministic call. What self-consistency recovers at high k is damage chain-of-thought did on the way in (§6), not headroom above the direct-prompt baseline.

Determinism carries a confidence signal, but not one to lean on. Determinism is lower among wrong answers than right ones in all six cells (Fig. 3c), so disagreement carries information — but at k=0 roughly half the errors come with none, and group sizes of 9–30 make those intervals too wide to rely on (Table 11).

This mechanism is specific to our regime: the sampling component of a selector-free compute axis, on two tasks, at one tier, with 3–4 samples. It says nothing about Best-of-N with a trained personalized reward model, where the binding constraint is the selector (Zhang & He, 2026) rather than the diversity of the candidate pool. Nor does it separate diversity from information: we predict that raising the sampling temperature would add answer diversity without producing a resolvable gain at k=0, because the missing quantity there is user information, not sample diversity. We record this before testing it.

6 CONTROLS

Chain-of-thought at matched k buys nothing here, and at high k it costs. Four of eleven matched-k contrasts are resolvable (App. G) and *three of the four are harms*: LaMP-2 k=32 -0.0816 macro-F1 $[-0.1529, -0.0033]$, LaMP-3 k=32 -0.4872 MAE $[-0.8462, -0.1795]$ (that cell is n=39 at temperature 1.0, so it conflates chain-of-thought with sampling temperature), LaMP-5 k=32 -0.1646 ROUGE-1 $[-0.2158, -0.1122]$. The one gain is LaMP-5 k=0 ($+0.0649$ $[0.0340, 0.0967]$): with no user signal, thinking out loud helps on free-text titling — which is why §5 calls the high-k sampling gain recovery.

The attainable ceiling does not move detectably when the retriever changes — at the resolution we achieved. Swapping BM25 for recency retrieval at matched k produces **no resolvable difference in 12 of 12 contrasts** on LaMP-1 and LaMP-2 (largest point estimate -0.061 macro-F1 at LaMP-2 k=16, interval $[-0.127, 0.013]$). Testing the ceiling directly — the paired difference of $\max_k q$(recency) $-$ $\max_k q$(BM25) — gives $+0.0012$ $[-0.0993, 0.0874]$ accuracy on LaMP-1 (n=99; BM25 0.6224 vs. recency 0.6237) and -0.0215 $[-0.0803, 0.0406]$ macro-F1 on LaMP-2 (n=90; 0.4099 vs. 0.3883).

This licenses describing the bound in §4.2 as a property of the information available rather than an artefact of BM25 — and exactly that much: the interval half-widths are ± 0.09 and ± 0.06, so **a ceiling shift below roughly six to nine points is not excluded**, only two of four tasks and one tier were run with a second retriever, and the recency baseline is *positional* rather than chronological. App. M gives the full caveat.

Measurement losses. 29 refusals (all mid-tier, direct prompting) leave mid-tier cell sizes between 93 and 99, so that tier's LaMP-2 numbers rest on a mildly non-random subsample of unknown bias direction, uncorrected; 168 usage-limit aborts emptied one grid-A and one grid-C cell. Every loss is itemised in the Reproducibility Statement, per-cell n and parse rates in App. C.

7 A DECISION RULE KEYED ON OBSERVABLE SIGNAL

If an upgrade's value depends on how much user signal is present, an operator should be able to decide from the signal alone, before paying for any model call. The available quantity is $\mathrm{eff}_k = \min(k, |\text{profile}|)$; we fit the expected gain from upgrading the tier as a decaying function of it and upgrade when the observed signal falls below a threshold,

$$\hat{g}(k) = c\,(k+1)^{-d}, \qquad \text{upgrade} \iff \mathrm{eff}_k \le k^\star(\tau), \qquad k^\star(\tau) = (c/\tau)^{1/d} - 1,$$

with τ the smallest quality gain that justifies an upgrade, left free since we do not have per-token prices. On LaMP-3 (c=0.1668, d=0.8942) the rule beats a matched-budget random policy at every $\tau > 0$ — regret 0.0162 against 0.0245 MAE at τ=0.02 — and the random baseline buys exactly as many upgrades, so that ordering is information rather than budget. **It does not improve quality over always upgrading** (0.0162 against 0.0057): its value is the compute it *declines*, cutting the upgrade rate from 100% to 71% at τ=0.02 for 0.0106 MAE. **And it does not transfer.** On LaMP-2 the fitted gain is nearly flat (d=0.010) precisely because the true gain *grows* with k (§4.4), so $k^\star$ diverges and the rule degenerates to always-upgrade; on LaMP-1 the per-k gains are noise; on LaMP-5 every effect is under 0.05 ROUGE-1. Usable on rating prediction and on no other task here (App. J).

8 DISCUSSION AND LIMITATIONS

The task-dependence is the result, not the noise. It would be convenient to report a single sign for the cross-term; the measurement does not allow it. Model scale and user signal are substitutes on rating prediction (DiD 0.182 MAE, p=0.003) and complements on movie tagging (0.157 macro-F1, p=0.016), both resolvable at every $k \ge 4$, so any account of when test-time compute pays for personalization that ignores the task will be wrong on one of these two. What generalises is the *ordering* — signal moves the metric first, further and more often (39/71 vs. 10/82) — and the *contingency*: on LaMP-2 the larger tier's advantage exists only once signal is present, and self-consistency's gain only at k=32. Both say: check whether the larger tier actually leads without history before treating a tier upgrade as the lever, and do not expect a sampling budget to substitute for history the model lacks. App. E records, in advance of testing it, which kind of task we expect each behaviour on. Those predictions and the one in §5 are what a second benchmark, a ladder with measured parameter counts, and a context-compression arm would test.

Limitations. *Compute axis:* model tier plus a 3–4 sample budget over a frozen model, selector-free — no reasoning-token or thinking budget and no Best-of-N with a trained personalized reward model, so our results neither contradict nor can be used against Zhang & He (2026)'s reward-model-mediated scaling law; and the ladder's two model-scale steps are why the LaMP-3 exchange rate of 3.65 is an extrapolation beyond it. *Signal axis:* $k \le 32$ items retrieved by BM25 from one user's own profile — nothing about $k > 32$, cross-user signal, or fine-tuning on user data — and a retriever control covering two tasks, one tier, and a positional recency baseline (§6). *Resolution:* 99 items per cell, 39 in grid B, so many marginal contrasts are unresolvable; hence resolvable-effect *counts* rather than cell-by-cell verdicts, effects below roughly $\pm$0.06–0.09 quality units outside our resolution, no multiplicity adjustment, and per-stratum gains (App. K) reported as exploratory. *Family and benchmark:* one frozen family, three tiers, four LaMP tasks, no published-baseline comparison (§3).

Conclusion. The two directions of this surface are not interchangeable — eight of a user's own items capture at least 96% of what retrieval delivers, while at equal n only 10 of 82 compute contrasts resolve at all — and the cross-term between them is task-dependent in sign. For personalization under sparse per-user signal, the first question is not how much compute to spend; it is how much of the user is in the context.

REPRODUCIBILITY STATEMENT

Data. All four task subsets derive from public LaMP (Salemi et al., 2024b) dev splits, downloaded from `https://ciir.cs.umass.edu/downloads/LaMP/LaMP_{N}/{split}/{split}_questions.json` and the matching `{split}_outputs.json`. We used `LaMP_1/dev`, `LaMP_2/new/dev` (movie tagging; note that `LaMP_2/dev` is a different task, news categorization, with a different profile schema), `LaMP_3/dev` and `LaMP_5/dev`. Dev splits are used for scoring because LaMP withholds test golds for its leaderboard. Query and gold id sets were verified equal for all four tasks, and profile key sets were tallied over all 1,001,968 profile items in the four full dev splits. Subset construction: exclude gold-leakage items (the gold string appears verbatim in the same user's own profile: 5 on LaMP-1, 9 on LaMP-5) and one malformed LaMP-2 input; sort the eligible pool by `(profile_len, qid)`; cut at ranks $n/3$ and $2n/3$ to form equal-sized profile-length terciles; within each stratum sort by `qid` and draw 50 indices with `random.Random(0).sample`. That gives 150 queries per task; each grid cell runs 99 of them, 33 per stratum. The subsets are saved as self-contained parquet files carrying the full untruncated profile, so no reproduction needs to re-download the ≈993 MB of raw dev data.

Retrieval and prompting. Okapi BM25 with k_1=1.5, b=0.75, a fixed inline stopword list, a per-task retrievable-field map, and ties broken by original profile order (determinism verified by repeat-call equality); the BM25 query is the payload with the shared instruction boilerplate stripped, so term statistics are not dominated by the identical instruction prefix (or, on LaMP-2, by the 15-tag list). Individual profile items are truncated to 1,200 characters. The recency retriever takes the profile tail; see the caveat below. Prompt templates for both the `direct` and `cot` variants are reproduced verbatim in App. L.

Models and the compute axis — an important constraint. Both configured GPU hosts were unreachable for this work, so **every model call went through a hosted API**. The consequence for the paper's claims is substantive rather than cosmetic: the serving interface exposes no reasoning-token or thinking-budget parameter, so **the compute axis is model tier plus sample budget, and contains no reasoning-token budget at all**. Additionally, the `temperature` parameter is rejected by the newest model generation available to us, so every sampling-based condition (grid B, and the LaMP-3 grid-C k=32 cell) ran on the preceding generation at `temperature=1.0`; grid A is a single greedy sample. The three tiers are the small, mid and large members of one frozen model family; we report them as Haiku / Sonnet / Opus tiers and give exact model identifiers in App. A. One caveat on the compute axis follows from that: the tier ordering we treat as a "ladder with two model-scale steps" is the provider's published tier ordering, and we have *no* parameter counts or FLOP counts for the three tiers. The exchange rate of §4.3 is therefore denominated in rungs of this ladder, not in a measured parameter or compute multiple; "3.65 model-scale steps on a ladder that has two" should be read as "no position on this ladder reaches the signal gain", not as a scale factor.

Metrics. MAE, RMSE, accuracy, macro-F1, ROUGE-1 and ROUGE-L are implemented in a standard-library-only harness with a 109-assertion test suite in which every metric is checked against a hand-computed case. ROUGE was cross-validated against Google's `rouge-score` (`use_stemmer=False`) on 307 cases and 1,842 individual comparisons with maximum absolute deviation 0. Predictions that cannot be parsed return `None` and are reported per cell and excluded from metric averages, never scored as wrong. LaMP-2 macro-F1 uses the closed 15-tag label set (App. B); every other metric is the harness's own output, and the item-level decompositions used inside the bootstraps were checked against it across 107 cells and 392 metric comparisons (maximum deviation 3.3×10^{-16}).

Inference settings. Percentile bootstrap over items, **2000 resamples, seed 12345**, paired: a resample draws query ids once and both cells of a contrast are re-scored on that same draw, restricted to the intersection of ids present in both cells. Functional-form comparison: paired repeated 5-fold cross-validation over items, 10 repeats (50 folds), seeds $12345 + r$, identical folds for every form. Decision rule: 20 random 50/50 splits of users, seeds 1000–1019, with 200 random-policy draws per split. Knee criterion: smallest k at which the running maximum of the quality curve reaches 90% of the total k=0→32 gain, with tiers pooled before estimation.

Measurement losses, stated completely. Of 11,394 responses: **29 refusals** (all mid tier, `direct` variant, `stop_reason = refusal`: 8 LaMP-1, 16 LaMP-2, 4 LaMP-3, 1 LaMP-5), non-uniform in k, so mid-tier cell sizes range from 93 to 99; **168 usage-limit aborts**, an infrastructure loss with no model content, which emptied the LaMP-5 / small tier / k=4 cell of grid A and reduced the LaMP-3 k=32 cell of grid C to 11 of 99 responses, which we treat as uninformative and exclude; and **14 unparseable responses of the 11,197 usable calls** (0.13%), 13 on LaMP-1 (6 of them in the recency k=32 cell, a 6.1% rate there) and 1 on LaMP-5. Per-cell n, parse failures and parse-failure rates for all 106 cells are in App. C. We make no correction for the refusal-induced subsampling and flag its direction as unknown.

A consolidation defect found after the analysis was complete. Three grid-C cells (LaMP-3 k=0 and k=4, LaMP-2 k=4) enter our analysis at n=60. This is *not* a measurement loss: all 99 responses were collected and are present in the released per-cell records. Grids B and C differ in no coordinate except the sample budget (Table 1), and the consolidation our analysis consumed de-duplicated on a key that omitted exactly that coordinate: for items present in both, the chain-of-thought control (one greedy sample) and the sampling grid (3–4 samples at temperature 1.0) collided at `sample_idx`=0 and one row was discarded. We ship both consolidations: `sweep_all_raw.parquet`, the exact input our analysis consumed, and `sweep_all_rebuilt.parquet`, rebuilt with the sample budget in the key. Of the 106 published cells the corrected table reproduces 89 exactly and differs on 17 — the three above, where the corrected values are *worse* for chain-of-thought (LaMP-3 k=4 MAE $0.1833 \rightarrow 0.2929$, LaMP-3 k=0 $0.3500 \rightarrow 0.4040$, LaMP-2 k=4 macro-F1 $0.4122 \rightarrow 0.4370$), and 14 grid-A cells at identical n and at most 0.0379 apart, where duplicate successful responses exist and the two consolidations kept different ones. **Every reported metric, interval and fit in this paper is computed on the uncorrected consolidation**, which is internally consistent and released; we did not re-run the analysis. One exception is worth stating precisely, because it sharpens rather than weakens the point above: the direct-prompting series of Figure 3(d) is scored on the 39 grid-B items, and on the uncorrected consolidation two of its six cells (LaMP-2 k=4 and k=32) retain 0 and 1 usable rows of 39 — that consolidation kept errored rows over the backfilled retries for cells that were partly re-run. That series is therefore computed on the corrected table, where all six cells are complete. No other figure or number depends on the correction. The substitution result is unaffected — it rests on LaMP-3 grid A, where no cell differs. The corrected data would move the chain-of-thought control of §6 *further* in the direction we report, not against it. A full per-cell diff is reproducible with `scripts/score_sweep.py --raw results/sweep_all_rebuilt.parquet --check`.

We record the cause rather than only the consequence, because it is the generalisable part. The defect is definitional, not incidental: the sample budget belongs to a cell's identity (§3), a key that drops it silently merges two experimental conditions that are otherwise indistinguishable, and the resulting shortfall presents as missing data rather than as an error. We first attributed these three cells to the usage-limit aborts, and only re-examining the per-cell records showed the responses had been collected. Any grid that varies a sampling parameter alongside a prompting variant has this failure mode; the defence is to make the cell key explicit in the write-up, not only in the code.

Known unverified assumptions. Two, both recorded rather than resolved. (i) The recency retriever assumes profile list order is chronological: LaMP profiles on these four tasks carry no date field, so "recency" is *positional*. This is standard in this literature but we could not verify it from the data, and it bounds what the retriever control in §6 establishes. (ii) We did not verify which LaMP-2 variant any published baseline used, which is why no number here is tabled against a published baseline.

Artifacts. Every number in this paper is generated from saved result files rather than transcribed: the per-cell metric table (106 cells), the 294 paired contrasts, the fit and cross-validation records, the decision-rule splits, the entropy tables and the control contrasts. All tables in the paper are emitted programmatically from those files by a single script, so a table cell and its source row cannot drift. We will release the harness, the four evaluation subsets, the raw per-call sweep records, the analysis code and the table/figure generators. A register of every claim in this paper not traceable to one of those files accompanies the release. One gap in that release is worth naming here: the per-cell metric table is regenerable from the saved responses by a released script, but the contrast code that produced the 294 bootstrap intervals was run inline and not saved, so the released `contrasts.csv` is a

record of those intervals rather than a reproduction path for them. The bootstrap is fully specified above (2000 resamples, seed 12345, paired, percentile, restricted to the id intersection) and we include a reimplementation from that specification, gated on reproducing the shipped intervals; we mark it as a reimplementation and not as the code that produced the paper.

ETHICS STATEMENT

This work studies how much of a user's personal history a language model needs in order to personalize its output. That is a privacy question as much as an accuracy one, and our results speak to both sides of it.

Data provenance and new collection. We introduce no new user data. All experiments run on the public LaMP benchmark (Salemi et al., 2024b), whose profiles are derived from public sources, downloaded from its published URLs and used only for evaluation. We collect nothing from human subjects, and we release only seeded evaluation subsets of already-public records together with model outputs on them. We did, however, find and remove a leakage defect (the gold answer appearing verbatim in the same user's own profile) — a reminder that even public personalization corpora need auditing before they are used to make claims about how much user history helps.

The privacy dimension is the point, not a side effect. A personalization system's appetite for user history is usually treated as unbounded: retrieve more, do better. Our measurements do not support that. On every task we could resolve, at least 96% of the achievable gain is reached by $k{=}8$ retrieved items, and the knee sits there on all three, though only two of those intervals localise it. The practical reading is a privacy-favourable one: a deployment that retrieves eight of a user's own items forgoes little measured quality relative to one that retrieves thirty-two, and a system designer who wants a smaller per-user retention or context footprint has a measured basis for choosing one. We would rather this result be used that way than as an argument for collecting more history. We caution against the opposite reading of our substitution result: "retrieve the user's history instead of buying a bigger model" is a cost-saving argument that also increases the amount of personal data placed in a model's context, and the two considerations trade against each other.

Cold start is an equity problem. The flip side of a signal-bound regime is that users with little history get less benefit — structurally, not incidentally. Our LaMP-2 measurements make this concrete: at a nominal budget of $k{=}32$, 60% of those users have shorter profiles than the budget and the mean number of items actually reaching the context is 22.1. Those users sit further left on a curve whose steepest region is exactly where they are. If quality is bounded by available user information, then in any deployment with sparse users — new accounts, infrequent users, users who decline tracking, users whose history was deleted — the service is systematically worse for them, and buying more inference compute does not close the gap on three of the four tasks we measured. We did not test any cold-start remedy, and we deliberately do not report per-stratum comparisons as results, because at $n{=}33$ per stratum they are within noise (§4, App. K). Fairness across history-length groups is a question this design can pose but not settle, and methods that borrow signal across users (Zhang et al., 2026) or elicit preferences online (Qu et al., 2025) raise their own privacy trade-offs.

Other considerations. We report a task-dependent result rather than a single sign for the cross-term, and we flag the tasks where our claims do not hold (§4.4), because a one-directional claim would be more quotable and less true. Model outputs on personalization tasks can encode or amplify user-level biases; we measured aggregate quality only and did not audit generated titles or tags for harmful content. No human subjects were involved and no IRB review was required.

AI USE STATEMENT

Every number in this paper was emitted programmatically from saved result files; none was retyped into LaTeX by hand. The metric implementations were checked against 109 hand-computed assertions, ROUGE against a reference implementation on 307 cases with zero deviation, and the item-level decompositions used inside every bootstrap against the harness scorer across 107 cells and 392 metric comparisons (maximum deviation 3.3×10^{-16}). All 40 bibliography entries were retrieved from live APIs rather than recalled. We state this first because the verification is what makes the disclosure below safe to give in full.

This disclosure is reconstructed from per-phase logs kept while the work was being done, against the category list on the ICLR 2027 AI Policy for Authors page as retrieved from `iclr.cc` during this work. Generative AI carried out essentially all of the work reported here — hypothesis refinement, experiment design, implementation, data cleaning, statistical analysis, interpretation, all four figures, and all of the prose — under human direction and constraint. The human authors set the research question and the selector-free constraint that defines the regime, and accepted the two findings that ran against their initial expectations: convergence to a common floor holds on one task of four rather than generally, and the interaction reverses sign on LaMP-2. They take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Required-disclosure categories, one by one. We used generative AI tools to *help develop conceptual frameworks*: the operational definitions that make the paper's thesis testable were AI-formulated, specifically the cross-term as a difference-in-differences over four cells sharing a query-id set together with the sign convention needed to read it as substitution rather than complementarity, the model-tier spread as the operational form of "convergence to a common floor", and the exchange rate as the operational form of "the compute axis cannot reach what the signal axis reaches". To *formulate mathematical claims*: the six candidate functional forms in §4.2, including the shared-floor restriction introduced specifically to turn an eyeball claim into a testable one, and the closed-form threshold $k^\star(\tau) = (c/\tau)^{1/d} - 1$ of §7. To *propose or refine hypotheses*: the central hypothesis was supplied by the human authors as "quality is data-bound, not compute-bound"; AI narrowed it to the signal $\times$ compute cross-term in the selector-free regime after the literature phase found the bare compute-marginal claim preempted by Zhang & He (2026), then *overturned* two of the human-supplied expectations against the data — convergence to a common floor holds on one task of four, and the interaction reverses sign on LaMP-2 — and the resulting task-dependence was made an explicit finding rather than suppressed. To *design and give feedback on research methodology and experiments*: the grid itself (the k values, the two compute components, the chain-of-thought and retriever controls), the paired bootstrap as the primary inference procedure, tier pooling before knee estimation, paired rather than unpaired cross-validation for form comparison, restriction of every cross-cell comparison to shared query ids, keying the decision rule on eff_k rather than k, and the matched-budget random baseline that tests whether the rule carries information. To *implement methods*: all code, without exception — the retrieval and metric harness, the sweep drivers, the fitting, bootstrap, contrast, decision-rule and entropy code, the figure code and the table generators. To *clean and reformat dataset*: the LaMP downloads, the schema inspection that corrected an initial wrong guess about the LaMP-2 profile schema, the leakage and malformed-input exclusions, the seeded stratified subsetting, the separation of the raw sweep records into grids A/B/C/R, the classification of API errors into refusals and usage-limit aborts, and the reconciliation of prompt-token counts across tiers via cached-read tokens. To *support qualitative and thematic data analysis*, in a limited sense: the classification of free-text API error strings into the two failure classes reported in §6, and the screening of retrieved abstracts during the literature phase. To *interpret results*: the verdict on each claim, the scoping of every claim to the tasks on which it is resolvable, and an explicit internal list of sentences the data does not support, which was used as a constraint while drafting.

We have **not** used generative AI tools to *generate synthetic data sets* — every number here comes from real LaMP data and real model calls — and the remaining required categories are **not applicable** to this work: *provide critical ingredients for proving mathematical claims* and *assist in the writing of proofs* (this is an empirical paper with no theorems), and *assist with translation* (the paper was drafted in English throughout).

Recommended-disclosure categories. Additionally, we used generative AI tools to *create or modify scientific figures* (all four, including two rounds of correction in which three panel titles were withdrawn after checking them against every plotted row); *suggest experimental parameters* (the k grid, BM25 hyperparameters, prompt truncation length, 2000 bootstrap resamples with seed 12345, the 90% knee criterion, the τ grid, and all random seeds); *create or edit software code* and *create artifacts* (the evaluation subsets, the harness, every result table and figure, and this manuscript's LaTeX source); *draft parts of a research paper* — in fact all of it; *summarize and analyse existing literature*, *identify relevant literature*, *discover research topics and identify gaps*, and *source and search for information* (30 OpenAlex queries and 55 arXiv queries yielding 618 and 882 unique records, 36 of them intersection queries run specifically to locate the novelty gap, plus page-level reads of the two most relevant papers); *format references* (all 40 bibliography entries were generated pro-

grammatically from retrieved API records); *brainstorming* (the query vocabulary and the reframing discussion); *edit a research paper to improve readability*; *suggest a structure for a research paper*; and *propose a title*. On the title specifically: the working title was supplied by the human authors; AI judged that it overclaimed once the interaction was found to reverse on LaMP-2, and proposed the present title, which the human authors accepted. The remaining recommended categories — *formulate questions for surveys or interviews* and *transcribe recordings of research material* — are not applicable.

We have reviewed all AI-assisted work. Beyond the checks stated at the top, and again stated concretely rather than as an assurance: 15 landmark arXiv identifiers were re-fetched by exact id lookup and confirmed against their expected titles, and 34 of the 35 arXiv-sourced bibliography entries were matched against OpenAlex at title-similarity 1.0 (one entry has no OpenAlex record and is flagged as arXiv-only); the ICLR 2027 policy pages were read from `iclr.cc` directly; and four factual statements in an internal task brief were found not to match the data file and were corrected against the file rather than carried forward.

What that review did not cover. Two of the most load-bearing prior papers were read at page level rather than cover to cover. For one of them we have since read the formal statements in the source — Theorem 3.1 and Lemma 4.2 of Zhang & He (2026) — and corrected this paper's characterisation of the oracle ceiling accordingly, from "logarithmic" to $O(\sqrt{\ln N})$; the other is characterised from its abstract, introduction and method sections. The two unverified assumptions listed in the Reproducibility Statement were recorded rather than resolved; and a style-reference document named in our own internal brief was absent from the machine, so presentation was calibrated against the official ICLR template's conventions instead. A register of every claim in this paper not traceable to a saved result file accompanies the artifact release, and it names these items explicitly. We consider disclosing the boundary of the verification to be part of the disclosure.

REFERENCES

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. In *Proceedings of the National Academy of Sciences*, 2024. doi: 10.1073/pnas.2311878121. arXiv:2102.06701.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws, 2022.

Chengcheng Guo, Kuo Cai, Yu Zhou, Qiang Luo, Ruiming Tang, Han Li, Kun Gai, and Guorui Zhou. PROMISE: process reward models unlock Test-Time scaling laws in generative recommendations, 2026.

Seungwook Han, Idan Shenfeld, Akash Srivastava, Yoon Kim, and Pulkit Agrawal. Value augmented sampling for language model alignment and personalization, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal large language models, 2022.

Bojian Hou, Xiaolong Liu, Xiaoyi Liu, Jiaqi Xu, Yasmine Badr, Mengyue Hang, Sudhanshu Chanpuriya, Junqing Zhou, Yuhang Yang, Han Xu, Qiuling Suo, Laming Chen, Yuxi Hu, Jiasheng Zhang, Huaqing Xiong, Yuzhen Huang, Chao Chen, Yue Dong, Yi Yang, Shuo Chang, Xiaorui Gan, Wenlin Chen, Santanu Kolay, Darren Liu, Jade Nie, Chunzhi Yang, Ellie Wen, Jiyan Yang, and Huayu Li. Kunlun: Establishing scaling laws for Massive-Scale recommendation systems through unified architecture design, 2026.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Édouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

George Kour, Itay Nakash, Ateret Anaby-Tavor, and Michal Shmueli-Scheuer. Think again! the effect of Test-Time compute on preferences, opinions, and beliefs of large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Industry Track)*, 2025. doi: 10.18653/v1/2025.acl-industry.45. arXiv:2505.19621.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. LongLaMP: A benchmark for personalized long-form text generation, 2024.

Weijiang Lai, Beihong Jin, Jiongyan Zhang, Yiyuan Zheng, Jian Dong, Jia Cheng, Jun Lei, and Xingxing Wang. Exploring scaling laws of CTR model for online performance improvement. In *Proceedings of the ACM Conference on Recommender Systems*, 2025. doi: 10.1145/3705328.3748046. arXiv:2508.15326.

Langming Liu, Shilei Liu, Yujin Yuan, Yizhen Zhang, Bencheng Yan, Zhiyuan Zeng, Zihao Wang, Jiaqi Liu, Di Wang, Wenbo Su, Pengjie Wang, Jian Xu, and Bo Zheng. UQABench: evaluating user embedding for prompting LLMs in personalized question answering. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025a. doi: 10.1145/3711896.3737385. arXiv:2502.19178.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-Time scaling for generalist reward modeling, 2025b.

Linhao Luo, Thuy-Trang Vu, Van-Anh Nguyen, Junae Kim, Gholamreza Haffari, and Dinh Phung. MATO: multi-objective personalized alignment with test-time optimization for large language models, 2026.

Fuyuan Lyu, Zhentai Chen, Jingyan Jiang, Lingjie Li, Xing Tang, Xiuqiang He, and Xue Liu. Exploring test-time scaling via prediction merging on Large-Scale recommendation, 2025.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. CoT-Valve: Length-Compressible Chain-of-Thought tuning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2025. doi: 10.18653/v1/2025.acl-long.300. arXiv:2502.09601.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling Data-Constrained language models, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: simple test-time scaling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.1025. arXiv:2501.19393.

Zikun Qu, Min Zhang, Mingze Kong, Xiang Li, Zhiwei Shang, Zhiyong Wang, Yikun Ban, Shuang Qiu, Yao Shu, and Zhongxiang Dai. T-POP: Test-Time personalization with online preference feedback, 2025.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. In *Foundations and Trends® in Information Retrieval*, 2009. doi: 10.1561/1500000019.

Alireza Salemi and Hamed Zamani. Comparing Retrieval-Augmentation and Parameter-Efficient Fine-Tuning for Privacy-Preserving personalization of large language models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025. doi: 10.1145/3731120.3744595. arXiv:2409.09510.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024a. doi: 10.1145/3626772.3657783.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: when large language models meet personalization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024b. doi: 10.18653/v1/2024.acl-long.399. arXiv:2304.11406.

Yash Ganpat Sawant. Theory-Grounded evaluation exposes the authorship gap in LLM personalization, 2026.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time compute optimally can be more effective than scaling model parameters, 2024.

Jinyan Su, Jinpeng Zhou, Claire Cardie, and Wen Sun. CLIPer: tailoring diverse user preference via Classifier-Guided Inference-Time personalization, 2026.

Zhongxiang Sun, Yi Zhan, Chenglei Shen, Weijie Yu, Xiao Zhang, Ming He, and Jun Xu. When personalization misleads: Understanding and mitigating hallucinations in personalized LLMs. In *Findings of the Association for Computational Linguistics: ACL*, 2026. doi: 10.18653/v1/2026.findings-acl.395. arXiv:2601.11000.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized Parameter-Efficient fine-tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2024. doi: 10.18653/v1/2024.emnlp-main.372.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency improves chain of thought reasoning in language models, 2022.

Linhai Zhang and Yulan He. Test-Time personalization: A diagnostic framework and probabilistic fix for scaling failures, 2026.

Yangtian Zhang, Leyao Wang, Hiren Madhu, Ngoc Bui, Walter Roznyatovskiy, and Rex Ying. CoPersona: collaborative persona graphs for robust LLM personalization, 2026.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents. In *ACM Transactions on Information Systems*, 2025. doi: 10.1145/3748302. arXiv:2404.13501.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, S. Mitra, Nedim Lipka, Nesreen K. Ahmed, and Yu Wang. Personalization of large language models: A survey, 2024.

A MODELS AND GRIDS

The three tiers are `claude-haiku-4-5-20251001` (small), `claude-sonnet-4-5-20250929` (mid) and `claude-opus-4-5-20251101` (large), all accessed through a hosted API. The 4-5 generation was used throughout because the newer generation rejects the `temperature` parameter, which grid B requires. Grid A: direct prompting, BM25 retrieval, one greedy sample, $k \in \{0, 1, 2, 4, 8, 16, 32\} \times 3$ tiers $\times$ 4 tasks (83 of 84 cells non-empty). Grid B: chain-of-thought, `temperature=1.0`, 3–4 samples per item, small tier, LaMP-2 and LaMP-3 at $k \in \{0, 4, 32\}$, 39 items per cell with a uniform sample count within each cell; the 39-item LaMP-3 subset is stratum-balanced (13/13/13) and its direct-prompt MAE tracks the full 99 items at k=4 (0.2051 vs. 0.2020) though k=0 is harder on it (0.5385 vs. 0.4545). Grid C: chain-of-thought versus direct at matched $k \in \{0, 4, 32\}$, small tier. Grid R: recency versus BM25 at matched $k \in \{1, 2, 4, 8, 16, 32\}$, small tier, LaMP-1 and LaMP-2.

B THE LAMP-2 MACRO-F1 LABEL SET

The reference LaMP scorer computes macro-F1 over $\mathrm{sorted}(\mathrm{set}(\text{gold}) \cup \mathrm{set}(\text{pred}))$. That denominator moves with the predicted label set, which makes macro-F1 non-comparable across cells and unstable under bootstrap resampling. We compute it against the closed 15-tag set instead. The data confirm the set is closed on this split: 14 tags appear in gold, 15 in predictions, and there are no out-of-vocabulary predictions. On the point estimate the two conventions agree (LaMP-2, small tier, k=8: 0.3938 either way).

C THE FULL GRID-A SURFACE

Tables 3–6 give every grid-A cell with its 95% paired bootstrap interval and its exact n. Brackets are percentile bootstrap intervals over items; n varies below 99 where mid-tier refusals removed items.

Table 3: LaMP-1, accuracy (higher is better).

k	Haiku (small)	Sonnet (mid)	Opus (large)	mean eff. k
0	0.4444 [0.3434, 0.5455] n=99	0.4124 [0.3093, 0.5155] n=97	0.4242 [0.3232, 0.5152] n=99	0.00
1	0.4848 [0.3838, 0.5859] n=99	0.4490 [0.3469, 0.5408] n=98	0.4646 [0.3636, 0.5657] n=99	1.00
2	0.5152 [0.4141, 0.6162] n=99	0.5204 [0.4184, 0.6224] n=98	0.5354 [0.4343, 0.6263] n=99	2.00
4	0.5051 [0.4040, 0.5962] n=99	0.5918 [0.4997, 0.6837] n=98	0.5960 [0.4949, 0.6869] n=99	4.00
8	0.5657 [0.4646, 0.6667] n=99	0.6429 [0.5510, 0.7449] n=98	0.5859 [0.4848, 0.6768] n=99	8.00
16	0.6224 [0.5204, 0.7172] n=99	0.6122 [0.5204, 0.7041] n=98	0.6162 [0.5152, 0.7172] n=99	16.00
32	0.5960 [0.4949, 0.6970] n=99	0.6531 [0.5612, 0.7449] n=98	0.5657 [0.4646, 0.6566] n=99	32.00

Table 4: LaMP-2, macro-F1 over the closed 15-tag set (higher is better).

k	Haiku (small)	Sonnet (mid)	Opus (large)	mean eff. k
0	0.4200 [0.2833, 0.4992] n=99	0.3048 [0.2175, 0.3664] n=94	0.3208 [0.2351, 0.3879] n=99	0.00
1	0.3572 [0.2435, 0.4300] n=99	0.3714 [0.2694, 0.4405] n=95	0.3456 [0.2517, 0.4119] n=99	1.00
2	0.3814 [0.2654, 0.4613] n=90	0.4107 [0.3014, 0.4862] n=99	0.3966 [0.2958, 0.4744] n=99	2.00
4	0.3792 [0.2770, 0.4465] n=99	0.4358 [0.3233, 0.5084] n=93	0.4388 [0.3254, 0.5123] n=99	4.00
8	0.3938 [0.2841, 0.4721] n=99	0.4665 [0.3483, 0.5359] n=99	0.4724 [0.3564, 0.5458] n=99	7.84
16	0.4223 [0.3094, 0.5006] n=99	0.4756 [0.3600, 0.5450] n=99	0.4389 [0.3309, 0.5085] n=99	13.86
32	0.4037 [0.2870, 0.4778] n=99	0.4797 [0.3645, 0.5544] n=98	0.4613 [0.3441, 0.5392] n=99	22.08

Table 5: LaMP-3, MAE on the 1–5 rating scale (lower is better).

k	Haiku (small)	Sonnet (mid)	Opus (large)	mean eff. k
0	0.4545 [0.3333, 0.5758] n=99	0.2653 [0.1735, 0.3574] n=98	0.2828 [0.1919, 0.3838] n=99	0.00
1	0.2727 [0.1818, 0.3737] n=99	0.2121 [0.1313, 0.2932] n=99	0.2323 [0.1414, 0.3333] n=99	1.00
2	0.2626 [0.1717, 0.3737] n=99	0.1856 [0.1031, 0.2784] n=97	0.1919 [0.1111, 0.2929] n=99	2.00
4	0.2020 [0.1111, 0.3030] n=99	0.1414 [0.0707, 0.2222] n=99	0.1414 [0.0707, 0.2222] n=99	4.00
8	0.1414 [0.0707, 0.2222] n=99	0.1111 [0.0505, 0.1818] n=99	0.1313 [0.0606, 0.2121] n=99	8.00
16	0.1919 [0.1111, 0.2929] n=99	0.1212 [0.0606, 0.1919] n=99	0.1515 [0.0806, 0.2323] n=99	16.00
32	0.1414 [0.0707, 0.2222] n=99	0.1224 [0.0612, 0.1939] n=98	0.1515 [0.0808, 0.2323] n=99	32.00

Table 6: LaMP-5, ROUGE-1 F (higher is better). The small tier at $k{=}4$ is the cell lost to usage-limit aborts.

k	Haiku (small)	Sonnet (mid)	Opus (large)	mean eff. k
0	0.3919 [0.3538, 0.4323] $n{=}99$	0.4268 [0.3801, 0.4709] $n{=}99$	0.4449 [0.3966, 0.4936] $n{=}99$	0.00
1	0.4164 [0.3713, 0.4607] $n{=}99$	0.4329 [0.3880, 0.4783] $n{=}99$	0.4441 [0.3954, 0.4935] $n{=}99$	1.00
2	0.4256 [0.3815, 0.4681] $n{=}99$	0.4496 [0.4030, 0.4961] $n{=}99$	0.4592 [0.4121, 0.5073] $n{=}99$	2.00
4	—	0.4774 [0.4281, 0.5243] $n{=}99$	0.4581 [0.4115, 0.5051] $n{=}99$	4.00
8	0.4275 [0.3811, 0.4717] $n{=}99$	0.4449 [0.3946, 0.4925] $n{=}99$	0.4485 [0.3976, 0.4963] $n{=}99$	8.00
16	0.4226 [0.3785, 0.4656] $n{=}99$	0.4546 [0.4062, 0.5037] $n{=}99$	0.4321 [0.3840, 0.4821] $n{=}99$	16.00
32	0.4175 [0.3687, 0.4646] $n{=}99$	0.4721 [0.4231, 0.5199] $n{=}98$	0.4396 [0.3933, 0.4870] $n{=}99$	32.00

Table 7: Every cell across all four grids with a non-zero parse-failure count. Unparseable predictions are excluded from metric averages, never scored as wrong.

Grid	Task	k	retriever	variant	n	parse failures (rate)
A	LaMP-1	16	bm25	direct	99	1 (1.0%)
R	LaMP-1	1	recency	direct	99	2 (2.0%)
R	LaMP-1	16	recency	direct	99	1 (1.0%)
R	LaMP-1	32	recency	direct	99	6 (6.1%)
C	LaMP-1	4	bm25	cot	99	1 (1.0%)
C	LaMP-1	32	bm25	cot	99	2 (2.0%)
C	LaMP-5	32	bm25	cot	99	1 (1.0%)
total, all 106 cells						14 (0.13% of 11,197 usable calls)

D FUNCTIONAL-FORM COMPARISON AND THE FITTED FLOOR

Table 8: The forms compared in Table 9. All are restrictions of the full per-tier saturating fit (first row), which is the reference the entries of that table are measured against. A subscript m marks a parameter free per tier; an unsubscripted parameter is shared by all three. Each form is fitted to the 21 cell errors of a task (7 values of k × 3 tiers), which is why the per-tier free-floor fits are weakly identified.

Row in Table 9	$\mathrm{err}(k)$	params	what the restriction tests
(reference) full per-tier fit	$e_{\infty,m} + a_m(k{+}1)^{-b_m}$	9	—
no signal effect	e_m	3	does retrieval do anything?
no compute effect	$e_\infty + a(k{+}1)^{-b}$	3	does model tier do anything?
floor only	$e_{\infty,m} + a(k{+}1)^{-b}$	5	do tiers differ *only* in the floor?
power law, no floor	$a_m(k{+}1)^{-b_m}$	6	is there an irreducible floor at all?
exponential in k	$e_{\infty,m} + a_m e^{-k/\tau_m}$	9	power-law or exponential decay?
one floor shared by all tiers	$e_\infty + a_m(k{+}1)^{-b_m}$	7	do all tiers share one floor?

Table 9: Paired repeated 5-fold cross-validation over items (10 repeats, 50 folds, identical folds for every form). Entries are the change in held-out MSE relative to the full per-tier saturating fit, $\times 10^3$, with the paired t across folds in parentheses. **Positive means the restricted form predicts worse**, i.e. the dropped term was needed. Forms are defined in Table 8.

Restriction of the full fit	LaMP-1	LaMP-2	LaMP-3	LaMP-5
no signal effect	+3.69 (+6.3)	-0.35 (-1.4)	+5.01 (+11.3)	-0.02 (-0.6)
no compute effect	-0.58 (-5.1)	-0.56 (-4.2)	+0.92 (+3.8)	+0.03 (+0.7)
floor only	-0.15 (-2.9)	-0.27 (-2.8)	+0.26 (+2.3)	+0.03 (+2.1)
power law, no floor	+0.01 (+0.1)	-0.10 (-3.2)	+0.25 (+2.1)	-0.03 (-1.5)
exponential in k	-0.10 (-2.0)	+0.05 (+1.4)	+0.00 (+0.1)	+0.02 (+5.3)
one floor shared by all tiers	-0.05 (-1.2)	-0.08 (-4.5)	-0.15 (-4.4)	+0.03 (+1.9)

Table 10: LaMP-3 shared-floor fit, $\mathrm{err}(k) = e_\infty + a_m(k{+}1)^{-b_m}$, $n{=}95$ items, in-sample RMSE 0.0194 MAE. The three exponents are statistically indistinguishable; only the amplitude separates the tiers. We quote no exponent as a result: see §4.2.

Parameter	Haiku (small)	Sonnet (mid)	Opus (large)
amplitude a_m	0.3480 [0.2583, 0.4696]	0.1615 [0.0954, 0.2649]	0.1811 [0.1165, 0.2770]
exponent b_m	0.9002 [0.4061, 1.6268]	0.9792 [0.3260, 2.2563]	0.9046 [0.2834, 1.8139]
shared floor e_∞		0.1206 [0.0323, 0.1808]	

Per-tier free-floor fits are omitted deliberately. On LaMP-1, LaMP-2 and LaMP-5 they are under-identified — e_∞ intervals run as wide as $[0.00, 0.72]$ and exponent intervals reach the optimiser bounds — and per-tier non-parametric knees at $n \approx 99$ have intervals mostly spanning $[2, 32]$. Neither should be quoted for individual tiers.

E WHY THE TWO TASKS MAY DIFFER — CONJECTURE

One reading of *why* the two tasks differ is available in our design, and we flag it as conjecture. LaMP-3 is ordinal regression over five labels and eight of a user's own reviews essentially reveal their rating offset, which a small model can apply; LaMP-2 is 15-way classification whose profiles are the shortest in the suite, so $k{=}32$ delivers a mean of only 22.1 effective items while requiring the model to integrate a large, weakly-labelled tag context. Capacity plausibly binds on the second task and not the first. We did not design an experiment to test that, and we do not claim it. What we can do is record the prediction it implies, before testing it.

A prediction recorded in advance. The account above is a claim about the *signal span*: we expect it to be large where a user's history reveals a low-dimensional parameter the model can then apply — an ordinal rating offset, on LaMP-3 — and small where that history must instead be integrated into a large, weakly-labelled output space. The rest is arithmetic rather than prediction: the head-to-head of §4.3 is identically the span minus the largest tier's lead at $k{=}0$, so a large span on a task where that lead exists yields substitution and a small one does not. On the two tasks of our four where the lead exists at all, the two quantities fall as the account requires (LaMP-3: lead 0.172, span 0.313; LaMP-5: lead 0.053, span 0.036) — but the account was written after those numbers were seen, so that is consistency, not evidence. The test is a task screened by output-space structure *before* it is run, and we state the prediction here so that it can be one.

F ANSWER DETERMINISM, PER CELL

Table 11 gives the per-cell values behind §5.

Table 11: Answer determinism and the headroom for self-consistency (grid B). Fully deterministic = every sample returned the identical parsed answer. Entropy is normalised by $\log(\text{parsed samples})$. The self-consistency gain is the majority vote over 3–4 samples minus the mean single sample, in quality units (accuracy for LaMP-2, MAE for LaMP-3). $n{=}39$ items per cell.

Task	k	samples	fully deterministic	mean norm. entropy	max distinct	self-consistency gain
LaMP-2	0	3	0.5897 [0.4359, 0.7436]	0.2700 [0.1634, 0.3767]	3	0.0085 [-0.0513, 0.0684]
LaMP-2	4	4	0.5641 [0.4103, 0.7179]	0.2218 [0.1369, 0.3076]	3	0.0128 [-0.0514, 0.0833]
LaMP-2	32	3	0.3846 [0.2308, 0.5385]	0.4213 [0.3146, 0.5402]	3	0.0855 [0.0171, 0.1624]
LaMP-3	0	4	0.6667 [0.5128, 0.8205]	0.1425 [0.0776, 0.2042]	2	-0.0064 [-0.0705, 0.0577]
LaMP-3	4	4	0.5128 [0.3590, 0.6667]	0.2466 [0.1633, 0.3291]	3	0.0000 [-0.0962, 0.0833]
LaMP-3	32	4	0.5128 [0.3590, 0.6667]	0.2097 [0.1449, 0.2794]	2	0.2500 [0.0962, 0.4231]

G FULL CONTRAST TABLES

Every contrast quoted in §4.3, §4.4 and §6, cell by cell. Tables 12 and 13 cover the two axes and their cross-term; Figure 4 and Tables 14–15 cover the two control families. All use the same items, bootstrap and interval criterion, so the counts in §4.3 compare like with like.

AXIS DOMINANCE AND THE CROSS-TERM

Table 13 tabulates the head-to-head and resolvable-effect counts of §4.3; Table 12 the difference-in-differences and exchange rates of §4.4.

Table 12: The cross-term. Left: difference-in-differences $[M(\text{large}, 32) - M(\text{large}, 0)] - [M(\text{small}, 32) - M(\text{small}, 0)]$ on the shared query ids, in raw metric units (n=93–99). Because LaMP-3's metric is an error, a *positive* DiD there means the compute advantage *shrinks* as signal grows; on the other three tasks a positive DiD means it grows. Right: spans in quality units and the model-scale steps needed to match the signal span; the ladder has two.

Task	DiD at k=32 (metric units)		Exchange rate		
	Opus − Haiku	Sonnet − Haiku	signal span	compute span (k=0)	steps needed
LaMP-1	-0.0101 [-0.1515, 0.1111]	0.0722 [-0.0515, 0.1856]	0.1515	-0.0202	unreachable
LaMP-2	0.1568 [0.0224, 0.2490]	0.1668 [0.0486, 0.2522]	-0.0163	-0.0992	unreachable
LaMP-3	0.1818 [0.0606, 0.3232]	0.1856 [0.0825, 0.2990]	0.3131	0.1717	3.65
LaMP-5	-0.0309 [-0.0742, 0.0143]	0.0214 [-0.0243, 0.0673]	0.0256	0.0530	0.96

Table 13: Left: the smallest model at its best k against the largest model at k=0, in quality units, paired over n=99 items, with the bootstrap p. Right: how often each axis produces a resolvable effect — signal contrasts are k vs. k=0 at fixed tier, compute contrasts are tier vs. tier at fixed k. Both families use the same items, bootstrap and interval criterion.

Task	Metric	Small model + history vs. largest model, no history		Resolvable contrasts	
		advantage at best k	p	signal	compute
LaMP-1	accuracy	0.1982 [0.1010, 0.2993] (k=16)	0.0005	13/18	0/21
LaMP-2	macro-F1	0.1015 [0.0212, 0.1708] (k=16)	0.0060	8/18	3/21
LaMP-3	MAE	0.1414 [0.0606, 0.2222] (k=8)	0.0005	15/18	4/21
LaMP-5	ROUGE-1	-0.0174 [-0.0646, 0.0281] (k=8)	0.4650	3/17	3/19
All				**39/71** (55%)	**10/82** (12%)

CONTROLS IN FULL

Figure 4 and Tables 14–15 give the two control families of §6 cell by cell.

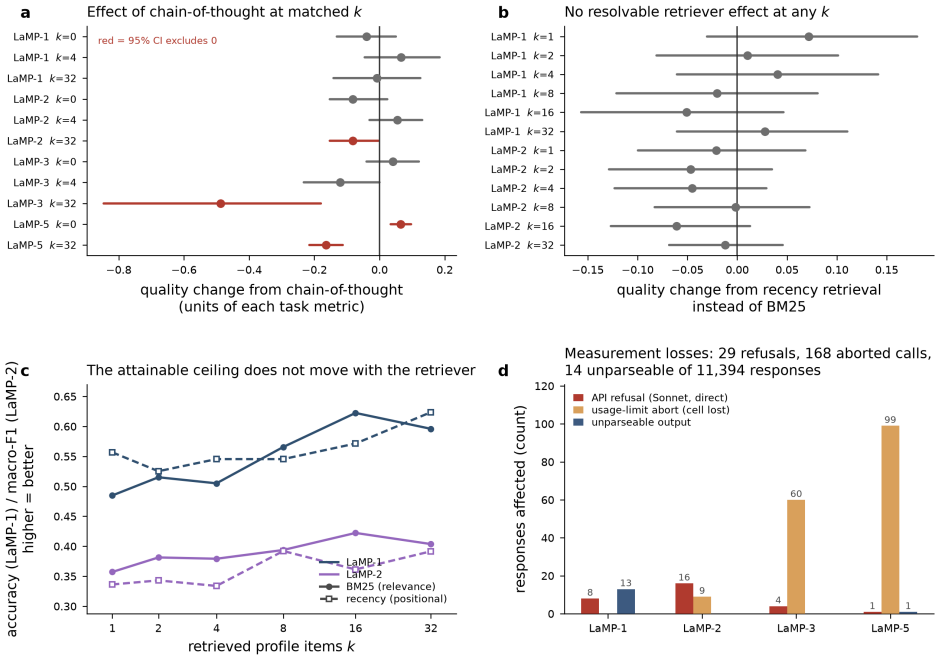

Figure 4: Controls. **(a)** Quality change from chain-of-thought at matched k (grid C, small tier); red intervals exclude zero. **(b)** Quality change from swapping BM25 for a positional recency retriever at matched k (grid R, small tier). **(c)** The two retrievers' curves and the best value each attains over k. **(d)** Measurement losses by task.

Table 14: Chain-of-thought versus direct prompting at matched k, small tier (grid C), in quality units. The LaMP-3 k=32 row is n=39 and at temperature 1.0, so it conflates chain-of-thought with sampling temperature; the other resolvable rows are n=99 and greedy.

Task	k	n	quality change from chain-of-thought	resolvable
LaMP-1	0	99	-0.0404 [-0.1313, 0.0505]	no
LaMP-1	4	99	0.0664 [-0.0461, 0.1842]	no
LaMP-1	32	99	-0.0083 [-0.1408, 0.1253]	no
LaMP-2	0	99	-0.0816 [-0.1524, 0.0238]	no
LaMP-2	4	99	0.0549 [-0.0310, 0.1312]	no
LaMP-2	32	99	-0.0816 [-0.1529, -0.0033]	yes
LaMP-3	0	99	0.0404 [-0.0404, 0.1212]	no
LaMP-3	4	99	-0.1212 [-0.2323, -0.0000]	no
LaMP-3	32	39	-0.4872 [-0.8462, -0.1795]	yes
LaMP-5	0	99	0.0649 [0.0340, 0.0967]	yes
LaMP-5	32	99	-0.1646 [-0.2158, -0.1122]	yes

Table 15: Recency versus BM25 retrieval at matched k, small tier (grid R), in quality units, plus the ceiling test $\max_k q(\text{recency}) - \max_k q(\text{BM25})$ over $k \in \{1, \ldots, 32\}$. No contrast resolves. Recency here is *positional*, not dated.

Task	k	n	quality change from recency instead of BM25	resolvable
LaMP-1	1	99	0.0719 [-0.0303, 0.1806]	no
LaMP-1	2	99	0.0101 [-0.0811, 0.1010]	no
LaMP-1	4	99	0.0404 [-0.0606, 0.1414]	no
LaMP-1	8	99	-0.0202 [-0.1212, 0.0808]	no
LaMP-1	16	99	-0.0510 [-0.1566, 0.0463]	no
LaMP-1	32	99	0.0277 [-0.0606, 0.1105]	no
LaMP-2	1	99	-0.0209 [-0.0996, 0.0685]	no
LaMP-2	2	90	-0.0468 [-0.1289, 0.0350]	no
LaMP-2	4	99	-0.0454 [-0.1230, 0.0293]	no
LaMP-2	8	99	-0.0017 [-0.0831, 0.0725]	no
LaMP-2	16	99	-0.0609 [-0.1270, 0.0127]	no
LaMP-2	32	99	-0.0121 [-0.0683, 0.0456]	no
LaMP-1	ceiling	99	0.0012 [-0.0993, 0.0874] (BM25 0.6224 vs. recency 0.6237)	no
LaMP-2	ceiling	90	-0.0215 [-0.0803, 0.0406] (BM25 0.4099 vs. recency 0.3883)	no

H LaMP-3 accuracy against the constant-prediction floor

The majority class on our 99-item LaMP-3 subset is rating 5 at accuracy 0.6364; on the full 2,500-item dev split the constant-prediction accuracy is 0.5912. Table 16 gives grid-A accuracy. Twenty of the 21 cells clear the subset floor. The single exception is the small tier at k=0 (0.6061): with no user signal the small model is worse than always answering "5". With eight retrieved reviews the same model reaches 0.8687. Every cell clears the full-split floor.

Table 16: LaMP-3 accuracy, grid A. Subset constant-prediction floor 0.6364.

k	Haiku	Sonnet	Opus
0	0.6061	0.7551	0.7273
1	0.7475	0.7980	0.7980
2	0.7677	0.8351	0.8384
4	0.8283	0.8687	0.8687
8	0.8687	0.8889	0.8788
16	0.8384	0.8788	0.8687
32	0.8687	0.8776	0.8687

I Available signal versus retrieval budget

Only LaMP-2 binds. Averaged over the three tiers, its mean $\text{eff}_k = \min(k, |\text{profile}|)$ is 7.84 at k=8 (8% of users capped), 13.86 at k=16 (38% capped) and 22.11 at k=32 (60% capped) rather than the nominal budget. Two bookkeeping notes, recorded rather than smoothed over. First, recomputing the k=32 figure as an unweighted mean over the three per-tier cells of our own cell table gives 22.08 rather than 22.11, because the cells have unequal n; both round to the 22.1 quoted in §3 and no claim depends on the difference. Second, our analysis record leaves LaMP-5 at k=4 blank, because the small-tier cell at that k is the one lost to a usage-limit abort (§6); the surviving mid- and large-tier cells at that k both give $\text{eff}_k = 4.00$ over n=99 items, so the blank reflects a missing model call and not missing profile information. Apart from that cell, on LaMP-1, LaMP-3 and LaMP-5 no user's profile is shorter than the budget at any $k \leq 32$, so $\text{eff}_k = k$ exactly. The per-cell values are the last column of Tables 3–6.

J Decision-rule gain fits and held-out evaluation

Table 18 gives the per-task gain fits and Table 17 the held-out evaluation on LaMP-3 discussed in §7.

Table 17: Held-out evaluation on LaMP-3 over 20 random 50/50 splits of users (seeds 1000–1019); c, d refitted on the training half of each split, all losses evaluated on the held-out half. Loss is mean per-item $|\hat{y} - y|$; regret is against a per-k oracle that sees held-out truth; the matched-budget random policy upgrades the same *number* of k buckets, chosen uniformly at random (200 draws per split).

τ	$k^\star(\tau)$ (median)	upgrade rate	held-out loss	regret vs. oracle	regret of matched-budget random
0.00	∞	100%	0.1982	0.0057	0.0057
0.01	35.7	82%	0.2045	0.0119	0.0182
0.02	13.8	71%	0.2088	0.0162	0.0245
0.05	3.7	49%	0.2152	0.0226	0.0356
0.10	0.8	22%	0.2263	0.0338	0.0487
always upgrade	—	100%	0.1982	0.0057	—
never upgrade	—	0%	0.2516	0.0591	—
oracle	—	—	0.1926	0.0000	—

Table 18: Gain model $\hat{g}(k) = c(k{+}1)^{-d}$ fitted on all items, with the implied thresholds. On LaMP-2 the fitted d is at its lower bound because the true gain *grows* with k, so $k^\star$ diverges and the rule degenerates to always-upgrade. Only the LaMP-3 fit is usable.

Task	c	d	$k^\star(\tau{=}0.02)$	$k^\star(\tau{=}0.05)$
LaMP-1	0.0119	0.1294	-1.0	-1.0
LaMP-2	0.0600	0.0100	diverges	diverges
LaMP-3	0.1668	0.8942	9.7	2.8
LaMP-5	0.0444	0.4407	5.1	-0.2

K PER-STRATUM GAINS — EXPLORATORY, NOT A RESULT

Table 19 reports the $k{=}0 \to 32$ gain within each profile-length stratum. We include it for completeness and explicitly *not* as a finding: $n \leq 33$ per group, the intervals are correspondingly wide, and the direction reverses across tasks (on LaMP-1 the long stratum gains most, on LaMP-3 the short one does). No claim in this paper rests on a stratum-level comparison.

Table 19: $k{=}0 \to 32$ gain in quality units by profile-length stratum and tier, $n \leq 33$ per group. Exploratory.

Stratum	Tier	LaMP-1	LaMP-2	LaMP-3	LaMP-5
short	haiku	-0.0303 [-0.1818, 0.1212]	0.0935 [-0.0091, 0.1604]	0.4848 [0.1818, 0.8182]	0.0562 [-0.0093, 0.1178]
	sonnet	0.1290 [-0.0323, 0.2903]	0.0938 [-0.0160, 0.1657]	0.2812 [0.1250, 0.4688]	0.0809 [0.0111, 0.1585]
	opus	0.1212 [-0.0303, 0.2424]	0.0766 [-0.0172, 0.1425]	0.0909 [-0.0909, 0.2727]	0.0171 [-0.0525, 0.0940]
medium	haiku	0.1818 [0.0303, 0.3636]	0.0106 [-0.0800, 0.1292]	0.2424 [0.0606, 0.4242]	0.0062 [-0.0560, 0.0682]
	sonnet	0.1818 [0.0000, 0.3636]	0.1741 [0.0549, 0.2648]	0.0625 [-0.0000, 0.1562]	0.0203 [-0.0652, 0.1098]
	opus	0.0303 [-0.0606, 0.1212]	0.1645 [0.0215, 0.2350]	0.1212 [0.0303, 0.2424]	-0.0373 [-0.1001, 0.0176]
long	haiku	0.3030 [0.0902, 0.5152]	-0.0795 [-0.1511, -0.0034]	0.2121 [0.0909, 0.3636]	0.0144 [-0.0527, 0.0798]
	sonnet	0.3939 [0.2121, 0.5758]	0.1289 [-0.0454, 0.2029]	0.0909 [-0.0303, 0.2121]	0.0328 [-0.0328, 0.0956]
	opus	0.2727 [0.0909, 0.4545]	0.1194 [0.0111, 0.1949]	0.1818 [0.0606, 0.3333]	0.0041 [-0.0639, 0.0691]

L PROMPT TEMPLATES

Both variants are built by the same function; the only difference is the final instruction line. With $k > 0$ retrieved items the prompt is:

```
Here are {K} items from this user's history, most relevant
first:
1.  {item}
...
{K}.  {item}

Now complete the following task for the SAME user.
```

{LaMP input field, verbatim}

{LaMP-2 only: ``Allowed tags: ...'' listing all 15 tags}

{variant instruction}

At k=0 the first block is replaced by the single line `No history is available for this user. Complete the following task.` Retrieved items are rendered per task: LaMP-3 as `N. [rating {score}] {text}`, LaMP-2 as `N. [tag: {tag}] description: {description}`, LaMP-1 and LaMP-5 as `N. title: {title} | abstract: {abstract}`, each field truncated to 1,200 characters.

The `direct` instruction lines are: LaMP-1 `Respond with exactly '[1]' or '[2]' and nothing else.`; LaMP-2 `Respond with exactly one tag from the list and nothing else.`; LaMP-3 `Respond with a single integer from 1 to 5 and nothing else.`; LaMP-5 `Respond with only the title, on one line, and nothing else.`

The `cot` instruction lines are: LaMP-1 `Think step by step about which reference matches this author's own citing habits, then give your final answer on the last line in the exact form 'Answer: [1]' or 'Answer: [2]'.`; LaMP-2 `Think step by step about how this user tags movies, then give your final answer on the last line in the exact form 'Answer: <tag>'.`; LaMP-3 `Think step by step about how harshly or generously this user rates, then give your final answer on the last line in the exact form 'Answer: <integer 1-5>'.`; LaMP-5 `Think step by step about this author's titling style, then give the final title on the last line in the exact form 'Answer: <title>'.`

The parser takes the text after the last `Answer:`/`Final answer:` marker, tolerating markdown and bullet decoration around it, else the last non-empty line, then validates against the task's label space and returns `None` on failure. A markdown-bold answer marker (`**Answer: <title>**`) leaking into the scored ROUGE string was a real defect found in live smoke testing and fixed with regression assertions before the sweep ran; on a free-text metric that failure mode is silent, so we record it here.

M RETRIEVER CONTROL: THE FULL CAVEAT

Section 6 reports that swapping BM25 for a recency retriever moves the attainable ceiling by no resolvable amount, and summarises the limits of that statement. In full: the interval half-widths are ± 0.09 (LaMP-1) and ± 0.06 (LaMP-2), so **a ceiling shift smaller than about six to nine points is not excluded**; only two of four tasks and one tier were run with a second retriever; and our recency baseline is *positional*, not chronological — LaMP profiles here carry no date field, so it takes the profile tail and assumes list order is chronological, an assumption standard in this literature that we could not verify from the data. A different retrieval *objective* — a dense retriever (Izacard et al., 2021), or a trained or diversity-aware one (Salemi et al., 2024a) — has not been tested and could move the ceiling. We do not claim the retriever does not matter; we report that at this resolution, on these two tasks, at this tier, it did not move the ceiling.