INERT PRIORS: AUDITING RELEVANCE-SELECTED MEMORY PRIORS AGAINST THEIR OWN MEASUREMENT FLOOR

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agent memory systems increasingly inject a handful of procedural priors—adapted from skills, past experiences or hand-written operations—into their retrieval planner, and choose those priors by relevance to the query. An earlier study reported that priors adapted from public agent skills lower judge accuracy relative to no priors, and called this a negative-transfer boundary. That priors-versus-no-priors comparison cannot separate three explanations: misaligned knowledge, an out-of-domain bank, or a lossy adapter. We first replace that comparison with one that can. It compares relevance-selected priors against priors drawn uniformly from the same bank, through the same adapter, at the same budget and injection point, so only the selection rule varies. On the earlier study's own outputs this contrast is larger and more stable than the one it reported. We then measure what that study never did: how far two executions of the identical configuration land apart. The earlier data contain such a pair, and its spread is the same size as every effect the study reported. Re-running the identified comparison on LongMemEval at $n{=}500$ with repeated independent executions per arm on three current backbones, we find the reported effect excluded on the one backbone with enough repeats to test it, and unsupported on two with fewer. On the backbone with the most repeats, a bootstrap over questions and executions bounds the relevance-versus-random difference within ± 1.65 points and excludes the 2.67 the earlier study reported. On two further backbones the difference is small and of opposite sign, though two runs per arm cannot exclude that size. Single runs of a selector temperature that interpolates between the two rules, an equal-budget raw prior format, an LLM re-ranker and a second benchmark scored without an LLM judge all land inside the same floor. The relevance ranking does degenerate—over a bank with nothing relevant in it, three of eighteen skills take half of all injected slots—but its cost is the relevance-versus-random contrast already bounded above. We report the audit, the floor and the prospective readings, and argue that results of this kind should be read against a measured floor, and a stated smallest effect of interest, before they are read at all.

1 INTRODUCTION

Long-term memory systems for LLM agents decide what to store, how to consolidate it, and what to retrieve when a question arrives (Wu et al., 2024; Maharana et al., 2024; Chhikara et al., 2025; Liu et al., 2026). A growing number of them condition the retrieval step on a small set of *priors*—short procedural cues drawn from a bank of skills, past experiences, or hand-written operations (Zhang et al., 2026; Fang et al., 2025; Yang et al., 2026)—and select those priors the way they select everything else: by relevance to the query. An earlier study reported that curated public skills, adapted into such priors, lowered a memory system's judge accuracy relative to a no-prior baseline, and called this a negative-transfer boundary. That finding leaves one question open. Did the drop belong to the skills being the wrong kind of knowledge, to the bank being the wrong domain, or to the adapter compressing the content away? This paper answers by first replacing the comparison, and then by measuring something the earlier work, and the literature it sits in, had not measured.

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.

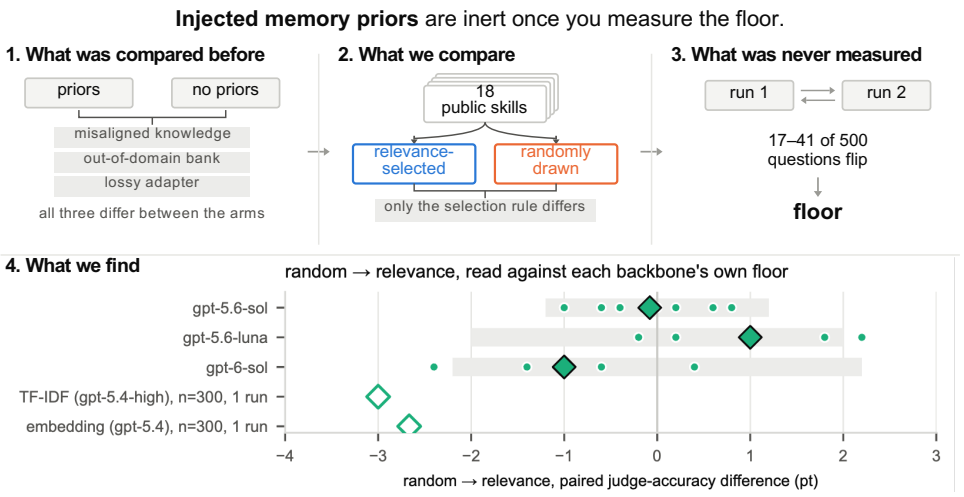


Figure 1: The argument in one figure. (1) The earlier priors-versus-no-priors comparison lets three explanations differ between its arms. (2) We compare relevance-selected priors against priors drawn uniformly from the same 18-skill bank, so only the selection rule differs. (3) We measure what was never measured: two executions of the same arm flip 17 to 41 of 500 questions across the three backbones, the floor. (4) Read against each backbone's own floor (grey band), RANDOM→RELEVANCE is -0.08, $+1.00$ and -1.00 points on three backbones; on the first the joint interval excludes the earlier size, on the other two it does not; the earlier single-run readings (-3.00, -2.67, hollow) had no floor at all.

An identified comparison. Priors-versus-no-priors cannot separate those three explanations, because all three differ between its arms. We compare instead two ways of choosing k priors from the *same* bank, through the same adapter, at the same token budget and injection point: rank by similarity to the query and take the top k, or draw k uniformly at random. Provenance, domain and compression are identical across the two arms; only the selection rule varies. On the earlier study's own outputs this contrast is -3.00 and -2.67 points across its two *instruments* (we use the word for a fixed selector, backbone and protocol), against -2.67 and $+0.00$ for the reported no-prior contrast. It is better identified and apparently more stable (Section 6.1).

A measured floor. A paired difference of a few points on several hundred questions means nothing until one knows how far two runs of the *same* configuration land apart. The earlier study's data contain such a pair; their paired difference reaches 1.67 points on 120 shared questions, the same order as every effect it reported (Section 5). We therefore re-run the comparison at $n{=}500$ with three independent executions per arm on a current backbone, and read every effect against that floor.

What we find. On `gpt-5.6-sol`, nothing moves (Figure 3). Relevance-selected, randomly drawn and absent priors give judge accuracies of 0.748, 0.749 and 0.753. The mean of every cross-arm contrast lies inside the 1.20-point within-arm floor, and a bootstrap over questions and executions bounds RANDOM→RELEVANCE within ± 1.65 points, excluding the 2.67 the earlier study reported. Single runs of a selector temperature that interpolates continuously between the two rules, of the raw skill text at the same budget, and of a second benchmark scored without an LLM judge land inside the same floor. The selector does degenerate as Section 4 describes, giving three of eighteen skills half the injected slots, but its cost is the relevance-versus-random contrast, bounded above within ± 1.65 points. A weaker backbone and a newer one give small differences of opposite sign, which two runs per arm constrain only loosely (Section 6.5).

Why this is the result and not a failed replication. The RAG-noise literature has shown how a reported gain can appear and vanish with prompt formulation, decoding limits and model generation (Cuconasu et al., 2024; Mazuryk et al., 2026). We apply that standard to the earlier claim, prospectively and with readings fixed in advance (Section 3), and it does not survive. The contribution is the audit: the comparison that identifies the selection rule, the floor that any such comparison has to

clear, and the finding that, at the scale where the floor is measured, the procedural priors this line of work has been injecting are inert to the resolution of a measured floor.

Contributions.

- An identified comparison for prior selection in agent memory—relevance-selected versus uniformly drawn priors from one bank—that holds provenance, domain and compression fixed.
- A repeat-measurement floor for LongMemEval-style evaluation, absent from prior work on this system, and the observation that the earlier effects were of the same order as it.
- A prospective test at $n{=}500$ with repeated executions of the selection-rule contrast on three backbones, and single-run checks of selector temperature, prior format, selector strength and evaluation protocol; no arm shows the reported effect, and where the repeats suffice an effect of that size is excluded.
- A characterisation of what a relevance ranking does over a bank with nothing relevant in it, and evidence that on a current backbone this degeneracy has no cost measurable above the floor.

2 Related Work

Memory systems and how they are evaluated. Long-term memory for LLM agents is now a crowded design space: Mem0 extracts and consolidates facts with an optional graph store (Chhikara et al., 2025), A-Mem keeps a self-evolving network of notes (Xu et al., 2025), MemoryOS and Zep organise memory by tiers and time (Kang et al., 2025; Rasmussen et al., 2025), and SimpleMem, our substrate, combines structured compression at write time with intent-aware retrieval planning at read time (Liu et al., 2026). LongMemEval (Wu et al., 2024) and LoCoMo (Maharana et al., 2024) are the two benchmarks these systems report on. Two recent audits of this evaluation practice bear directly on our design. MemDelta varies one component at a time on LongMemEval-S and finds that swapping only the embedding model in an otherwise identical pipeline shifts accuracy by 6.2 points at $n{=}500$, enough to reverse a system ranking (Wang, 2026). Our comparison therefore fixes the embedding model, the write-side store, the backbone and the judge across arms and lets one thing vary.

Procedural priors from skills. Public agent ecosystems distribute skills as reusable procedures, and SkillsBench reports that a few curated skills raise agent pass rates where self-written ones do not (Li et al., 2026). MemSkill reframes memory operations themselves as learnable skills selected by a controller (Zhang et al., 2026); Memp and related work treat procedural memory as a first-class target (Fang et al., 2025; Yang et al., 2026; 2025). The earlier study asked whether public execution skills, adapted into compact priors, transfer into a memory system's retrieval planner at test time without any learning, and reported a negative-transfer boundary. This paper replaces the comparison behind that claim with one that can be interpreted (Section 3).

Selection by relevance, and what happens when nothing is relevant. Retrieving by similarity to the query is the default in retrieval-augmented generation and in in-context example selection alike, and the literature on when it fails is instructive. Cuconasu et al. found that adding *random* documents to a RAG prompt raised accuracy by up to 35%, while the retriever's highest-scoring documents that did not contain the answer hurt (Cuconasu et al., 2024). Mazuryk et al. then reproduced that result under its original settings and showed it appears, weakens or disappears with small changes to prompt formulation, output-length limits and model generation. They attribute the original gain to a baseline that restrictive instructions had artificially handicapped (Mazuryk et al., 2026). Our comparison differs—the injected text is a procedural prior, not evidence, and the alternative to relevance is a uniform draw from a fixed bank rather than no retrieval—but the lesson transfers whole. We adopt their checklist prospectively: we vary the backbone, the selector, the judge and the evaluation protocol, and we include an arm that removes the adapter's compression so that the format of the injected text cannot be the explanation.

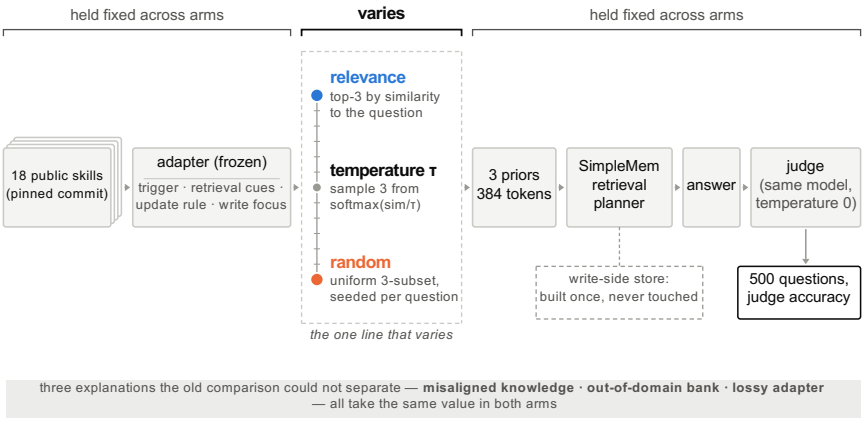


Figure 2: The identified comparison. Bank, adapter, budget, injection point, memory store, backbone and judge are held fixed across arms; the only line that varies is how the k=3 priors are chosen: top-3 by similarity (RELEVANCE), a uniform seeded 3-subset (RANDOM), or a temperature τ between the two (Section 4).

Selector degeneracy. That a similarity ranking over candidates none of which is relevant collapses onto a few generic items has, to our knowledge, no precedent in the memory literature; Section 4 places arg-max and uniform selection at the two ends of one axis and tests whether accuracy moves along it.

3 Setup and the Identified Comparison

Memory substrate and intervention point. We use SimpleMem (Liu et al., 2026) unchanged as the memory system. It builds a store from the interaction history at write time and, at read time, runs a retrieval planner that decides which slices of the store to search before an answer is generated. Our intervention touches only the planner: a short block of text, the *prior block*, is prepended to the planner's instructions. The store, the write-side pipeline, the answer generator, the backbone model and the judge are identical across every arm of every experiment. Reflection rounds are disabled throughout, matching the configuration of the earlier runs we compare against.

Bank, adapter and budget. The bank $\mathcal{B}$ is the 18 skills of a public agent-skill repository at a pinned commit, each adapted once by a fixed procedure A into a four-field prior $p = A(s)$: trigger, retrieval cues, update rule, write focus. The adapted text of a prior is what the selector matches against and what the planner receives. A prior block holds k=3 priors under a total budget of B=384 tokens, each prior truncated to $\lfloor B/k \rfloor$ tokens and the block to B. These choices are inherited from the earlier study and are held fixed here; they are not tuned.

The two arms. Given a question q, let $\mathrm{sim}(q, p)$ be the selector's similarity between the question and a prior's adapted text. The two arms differ in a single line:

$$\text{RELEVANCE}: \quad P_k(q) = \mathrm{TopK}_{p \in \mathcal{P}_\mathcal{B}} \mathrm{sim}(q, p),$$
$$\text{RANDOM}: \quad P_k(q) \sim \mathrm{Uniform}\big(k\text{-subsets of } \mathcal{P}_\mathcal{B}\big),$$

with the random draw seeded per question so that a run is reproducible and different seeds give different draws. Everything downstream of $P_k(q)$ is shared: the same formatter, the same budget, the same insertion point. We also keep the no-prior arm, BASE: the unmodified system, and the earlier study's reference.

Why this contrast is identified and the earlier one was not. The earlier result compared RELEVANCE with BASE. Three explanations for a drop were live and could not be separated: (A) execution skills are misaligned with what a memory planner needs; (B) this bank is from the wrong domain for a personal-memory benchmark; (C) the adapter's compression destroys whatever was

useful. All three are properties of *what text is available to inject*, and all three differ between the two arms of that comparison, because one arm injects nothing. Between RELEVANCE and RANDOM, by construction, the available text is the same set $\mathcal{P}_{\mathcal{B}}$, the adapter is the same A, and the domain is the same bank. (A), (B) and (C) take the same value in both arms. Only the map from a question to a k-subset differs, so any paired difference between the arms is attributable to the selection rule.

Selectors. We run two selector backends for RELEVANCE: TF–IDF over the adapted prior text, and a local sentence-embedding model (`all-MiniLM-L6-v2`) applied to the same text. MemDelta showed that the embedding model alone can move LongMemEval accuracy by several points (Wang, 2026). We therefore fix the embedding model to the one the memory store already uses and never vary it within a comparison. A third backend, an LLM re-ranker over the top-10 embedding candidates, is run as the strongest available selector so that the result cannot be attributed to a weak one.

Benchmarks and scoring. LongMemEval-S (Wu et al., 2024) has 500 questions over six abilities; we use all 500 rather than the 300-question subset of the earlier study, and score with the benchmark's LLM-as-judge protocol, with the judge model equal to the backbone within each run. LoCoMo (Maharana et al., 2024) is scored by token-level F1 against the reference with no LLM judge, which gives a second evaluation protocol with a different failure surface. The earlier study's numbers came from backbone models that are no longer served; we regenerate every number in this paper, and where we report the earlier runs we say so.

Readings fixed in advance. Before any new run was launched we wrote down how each result would be read; the plan was internal and not deposited externally, so we call it prospective rather than pre-registered. The selection effect is called present only if the paired difference between RANDOM and RELEVANCE is significant by an exact McNemar test at $n{=}500$ *and* its magnitude exceeds the largest paired difference observed between two runs of the same arm. Passing one test but not the other is reported as a direction without a size. Because that floor is an order statistic over few pairs, Section 5 adds the interval the inference rests on. The mechanism (Section 4) is called supported only if accuracy is monotone in the selector temperature across five levels. A negative reading on any arm is reported in the main text.

4 What the Selector Does When Nothing Is Relevant

There is no relevant prior to find. The bank is a repository of software-and-office skills—document co-authoring, web-app testing, algorithmic art, brand guidelines—and the benchmark asks about a user's own life across sessions. On the 300 questions of the earlier study the median top-1 similarity between a question and its best-matching adapted prior is 0.067, and 98.7% of questions have no prior above 0.15. A question about how long the user waited for an asylum decision selects `webapp-testing`, `doc-coauthoring` and `frontend-design`. Whatever the ranking is doing, it is not finding relevant guidance, because there is none.

The ranking concentrates. A natural expectation is that ranking noise-level scores behaves like a random draw. Instead it concentrates. Figure 4a shows, over the same 300 questions, how the two rules spread their $3{\times}300$ injected slots across the 18 skills. Relevance selection gives three skills half of all slots—one, `algorithmic-art`, takes 24.2% alone—and produces only 130 distinct three-prior sets, the most common of which is injected 16 times. Uniform selection uses the same 18 skills, gives the top three 20.2% of slots, produces 256 distinct sets, and repeats none more than three times. The planner receives irrelevant procedural text in both arms; under relevance selection it receives *the same* irrelevant text, question after question.

What the earlier data can and cannot say about it. The concentration is not one toxic skill (on questions where RELEVANCE loses to RANDOM, each skill's share matches its overall share) and does not track how weak the match is (median top-1 score 0.066 on lost questions, 0.058 on won ones). Whether this concentration *causes* the accuracy loss is a separate question the earlier data cannot answer: splitting the 300 questions by how often their prior set recurs gives a gradient in the predicted direction on 12 and 7 discordant pairs, neither distinguishable from zero. We therefore state the hypothesis and fix its test in advance.

The test: one dial between the two rules. Let $s_p = \mathrm{sim}(q, p)$ and draw k priors without replacement from $\mathrm{softmax}(s/\tau)$, implemented as Gumbel-top-k with a per-question seed. At τ=0 this is exactly RELEVANCE; as $\tau \to \infty$ it is exactly RANDOM; in between it is the same selector with a controlled amount of concentration. As an offline check of the dial alone, before any judged run: on the bank, with 200 seeds and one representative query, the top-1 skill's share of draws falls from 1.00 at τ=0 through 0.90, 0.47 and 0.23 at $\tau \in \{0.02, 0.05, 0.2\}$ to 0.22 at τ=10^6 (uniform: 0.17). If concentration is what costs accuracy, judge accuracy is monotone in τ across these five points; if it is not, the two endpoints differ for some other reason and the interior is flat or non-monotone. Section 6 reports which.

5 THE REPEAT-MEASUREMENT FLOOR

A paired accuracy difference between two arms is a number; whether it is a *finding* depends on how far apart two runs of the *same* arm land. Every earlier evaluation of this system reported the first quantity and never measured the second. We measure it two ways.

The floor that was already in the earlier data. Two of the earlier study's runs, at 120 and 300 questions, share an identical configuration and were executed ten hours apart on the same day; on the 120 questions they have in common they are a repeat measurement. Across the six arms of that study the paired difference between the two executions ranges from 0.00 to $\pm$1.67 points (Table 6, appendix). On those same 120 questions the effects the study reported—and the one we report—are -1.67 to -3.33 points: the same order as the floor, on a sample where neither is measurable. So the rest of this paper uses n=500 with repeated runs per arm.

The floor measured at the scale we report. For the new instrument we run every arm three times under one configuration, differing only in the run identifier, on all 500 questions, and read the paired difference between each pair of repeats exactly as we read the difference between arms. Figure 3 shows the within-arm pairs as grey points and Table 7 (appendix) lists them; every cross-arm difference in Section 6 is stated against the largest of them. On every backbone it comes from a deterministic arm (RELEVANCE on `gpt-5.6-sol` and `gpt-5.6-luna`, BASE on `gpt-6-sol`), not from the seed variance of RANDOM.

What the floor is, and how it is read. It is the empirical scatter of the whole pipeline—planner, generator and judge, all at temperature zero—between executions; a backbone served through a gateway is not deterministic, and the judge is a second such call. It is also an order statistic over few pairs (16, 3 and 3), so we use it as a scale, not as an inferential rule. The inference rests on a 95% interval from a bootstrap that resamples the questions and, within each arm, the executions (Appendix F); an effect of a given size is called excluded only when the interval excludes both signs of it. The smallest effect of interest is 2.67 points: the effect the earlier study reported under its TF–IDF instrument, and the smaller of the two relevance-versus-random readings on its outputs (Table 1).

6 RESULTS

6.1 THE EARLIER INSTRUMENTS: THE EFFECT THE STUDY REPORTED, AND THE ONE IT CONTAINED

Table 1 re-reads the earlier study's own item-level outputs. The effect it reported, BASE→RELEVANCE, shrinks from -2.67 points on the TF–IDF instrument to zero on the embedding instrument. The contrast this paper is built on, RANDOM→RELEVANCE, keeps its sign and size on both, and its TF–IDF bootstrap interval excludes zero.

6.2 THE NEW INSTRUMENT AT n=500: EVERY ARM INSIDE THE FLOOR

On `gpt-5.6-sol`, with three runs of BASE, three of RELEVANCE and five seeds of RANDOM, all on the full 500 questions, the three arms are indistinguishable from one another and from repeat noise (Figure 3b; Table 10 in the appendix). The 16 within-arm repeat pairs of Table 7 differ by up

Instrument	Contrast	Δ (pt)	95% CI	disc.	McNemar p
TF–IDF, gpt-5.4-high	BASE→RELEVANCE	-2.67	$[-5.33, +0.00]$	18	0.096
	RANDOM→RELEVANCE	-3.00	$[-6.00, -0.33]$	19	0.064
Embedding, gpt-5.4	BASE→RELEVANCE	$+0.00$	—	16	1.000
	RANDOM→RELEVANCE	-2.67	—	16	0.077

Table 1: Paired contrasts on the earlier study's 300 questions, from its own judge labels. CI: paired bootstrap, 20,000 resamples, the method the earlier study used. The effect the study reported, BASE→RELEVANCE, shrinks to zero on the second instrument; the selection contrast, RANDOM→RELEVANCE, keeps its size on both. Neither reaches $p < 0.05$.

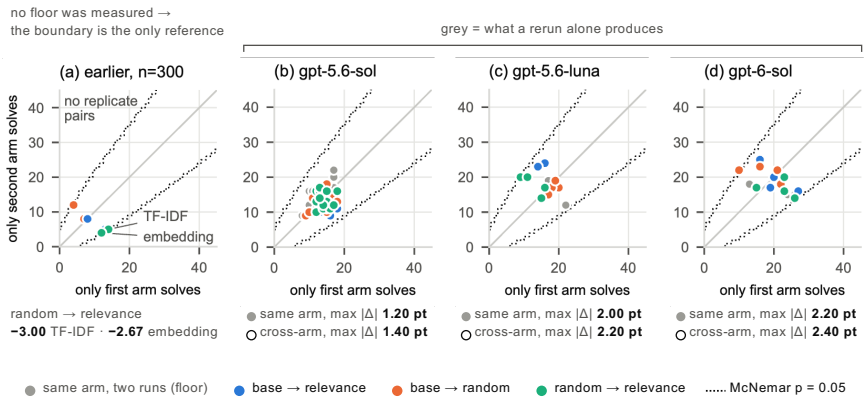


Figure 3: Each point is one pair of runs: x counts the questions only the first arm solves, y those only the second solves, so the diagonal is zero paired difference and the dotted staircase is exact McNemar p=0.05 at that n. Grey points are two executions of the *same* arm—what a rerun alone produces; they are the floor. **(a)** The earlier study's two instruments had no grey points; its RANDOM→RELEVANCE readings sit on the boundary. **(b–d)** On all three current backbones the coloured points sit in the grey cloud, at most 0.2 points beyond its widest pair. Data: the judge files in the supplement.

to 1.20 points. The 39 cross-arm pairs differ by up to 1.40 points, with a mean absolute difference of 0.56 against a mean within-arm $|\Delta|$ of 0.50. RANDOM→RELEVANCE averages -0.08 points over 15 run pairs, ranges from -1.00 to $+0.80$, and no single pair reaches $p < 0.44$. The joint interval is $[-1.65, +1.56]$ (Table 8): it excludes the 2.67 the earlier study reported and bounds any residual selection effect within 1.65 points. By the prospective rule the effect is absent; by the interval, it is at most a fraction of the one under audit.

6.3 THE SELECTOR TEMPERATURE: NO TREND IN SINGLE RUNS

The mechanism of Section 4 predicts that accuracy rises monotonically with τ as the injection decorrelates. It does not (Figure 4b,c). Mean judge accuracy over the five temperatures spans 0.748 to 0.754 (Table 5); each of the three interior temperatures was run once, and each lies within 0.80 points of τ=0 and inside the floor; the rank correlation of τ with accuracy over five levels is $+0.70$, short of the threshold fixed in advance. These are single runs, and by our own argument they bound nothing on their own; they say only that no interior temperature moved accuracy by more than the floor. The selector does concentrate exactly as described—that is a property of the ranking and is unchanged—but its cost is the RANDOM→RELEVANCE contrast of Section 6.2, bounded within 1.65 points on this backbone.

6.4 STRONGEST SELECTOR, SECOND PROTOCOL, UNCOMPRESSED PRIORS

Each row of Table 2 removes one alternative explanation for a null; each is a single run, and each lands inside the floor. Injecting the original `SKILL.md` text instead of the adapted four fields gives 0.748 under relevance selection and 0.738 under random selection ($+1.00$, $p = 0.44$). An equal-

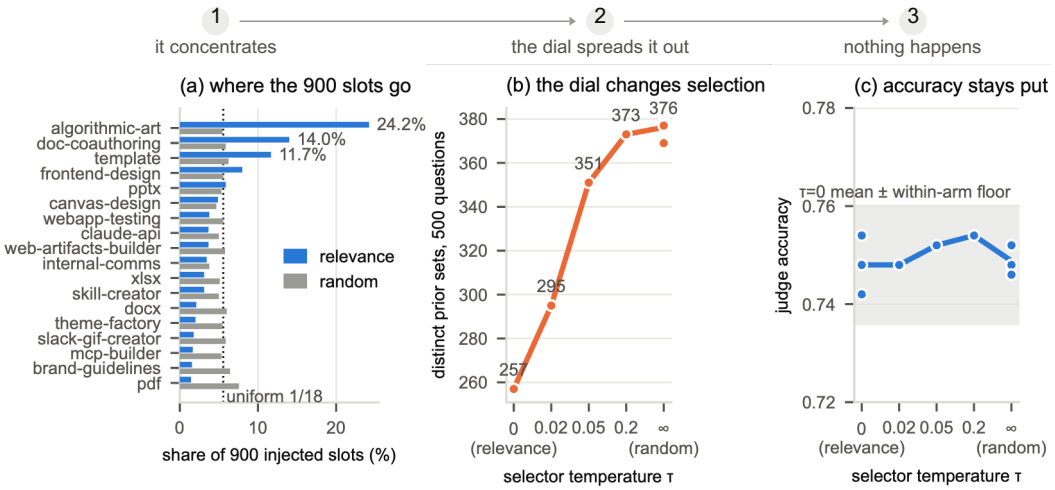


Figure 4: What the selector does and what it costs. **(a)** Share of the 900 injected slots per skill on the earlier instrument (TF–IDF, 300 questions × 3 slots): three skills take half, `algorithmic-art` alone 24.2%; a uniform draw gives each about 1/18. **(b)** Distinct three-prior sets actually injected over the 500 questions on `gpt-5.6-sol`, per run, along τ: the dial moves concentration by half. **(c)** Judge accuracy along the same τ, each run a point; the band is the τ=0 mean ± the within-arm floor. Accuracy never leaves the band. Table 5 in the appendix gives the numbers.

Arm	Run	Acc / F1	Δ (pt)	Disc.	p
rerank	`sol_rerank_curated_r1`	0.748	−0.40 (vs RANDOM)	30	0.856
raw-relevance	`sol_raw_curated_r1`	0.748	+1.00 (vs raw-RANDOM)	27	0.442
raw-random	`sol_raw_random_s1`	0.738	—	—	—
LoCoMo F1 BASE	`sol_locomo200_r1`	0.336	—	—	—
LoCoMo F1 RELEVANCE	`sol_locomo200_r1`	0.344	—	—	—
LoCoMo F1 RANDOM	`sol_locomo200_r1`	0.338	—	—	—
LoCoMo paired RANDOM→RELEVANCE		+0.52 pt	moved 41/200		

Table 2: Arms that each remove one alternative explanation: the strongest selector (LLM re-rank), the uncompressed prior format, and a second benchmark scored without an LLM judge.

budget raw prefix thus reveals no effect; at 128 tokens per prior it is still a compression, so it isolates the adapter from the truncation. On LoCoMo, scored by F1 without an LLM judge, BASE, RELEVANCE and RANDOM are 0.336, 0.344 and 0.338, with RANDOM→RELEVANCE at +0.52 points and an interval of $[-0.88, +1.95]$; only 41 of 200 questions change score at all. The strongest selector we can build—an LLM re-ranker, on the same backbone, over the top-10 embedding candidates, with a written justification per choice—gives 0.748, −0.40 points against RANDOM ($p = 0.86$). A selector that reads the question and reasons about the match, over the same ten candidates, does no better than one that never reads the question.

6.5 TWO MORE BACKBONES: NO EFFECT, WIDER INTERVALS

If the earlier gap were a property of weaker models that a stronger one masks, `gpt-5.6-luna`—whose no-prior accuracy is 0.727 against `sol`'s 0.753—should show it; and a newer model, `gpt-6-sol`, gives a third reading. Neither shows it, and with two runs per arm neither can exclude it (Figure 3c,d; Tables 11 and 12 in the appendix). On `luna`, two runs per arm give BASE 0.728/0.726, RELEVANCE 0.724/0.744, RANDOM 0.726/0.722. The two RELEVANCE runs differ by 2.00 points, the largest within-arm difference on this backbone and hence its floor. The twelve cross-arm pairs reach at most 2.20 (mean 0.80), and RANDOM→RELEVANCE averages +1.00 over four pairs (−0.20 to +2.20, pooled $p = 0.085$; joint interval $[-1.40, +3.30]$). Seven questions carry a failed answer in one luna run and are scored wrong (Appendix G); without them, a reading not fixed in advance, the contrast is +0.61 and the floor 2.23. On `gpt-6-sol`, BASE gives 0.718/0.740, RELEVANCE 0.736/0.718, RANDOM 0.732/0.742; its floor is 2.20, the cross-arm pairs

reach 2.40 (mean 1.17), and RANDOM→RELEVANCE averages -1.00 (-2.40 to $+0.40$, $p = 0.125$; joint interval $[-3.60, +1.50]$). The sign of that contrast is -0.08, $+1.00$ and -1.00 across the three backbones: the point estimates sit inside every floor and the direction does not repeat. But two runs per arm leave intervals that include ± 2.67, so on these two backbones the reported size is unsupported rather than excluded. The capability-conditional reading was one of two we fixed in advance; the data give it no support.

Taken together. On the backbone with enough repeats an effect of the reported size is excluded; on the other two the point estimates are small and of opposite sign, and the intervals are too wide to exclude it. The single-run arms land inside the floor and claim no more than that. Only one case ever showed an effect—single unfloored runs on two withdrawn backbones—and it is the one that cannot be re-checked. The effect was never shown to be more than noise, and where repeats let us check, its size is excluded.

7 LIMITATIONS

One memory system. Every arm runs on SimpleMem. Its no-prior accuracy on LongMemEval-S is 0.73–0.75 under our backbones; systems such as Mem0 report figures above 0.9 on the same benchmark (Chhikara et al., 2025). Our claim concerns the *selection rule* for injected priors, not absolute accuracy; whether relevance selection degenerates the same way inside a stronger substrate, or one whose planner is less exposed to injected text, is untested. A second substrate is the first thing we would add.

One bank. The 18-skill bank is small, from one repository, and out of domain for the benchmark by construction. The degeneracy we describe is a property of that mismatch. A bank with genuinely relevant priors would be testing a different question.

One budget. $k{=}3$ priors and $B{=}384$ tokens are the earlier study's values, held fixed to re-read its comparison. A null at this budget speaks only to the block this line of work injects.

Closed backbones. A commercial gateway serves all backbones at temperature zero without determinism across executions. The floor absorbs this, but every number here depends on models we do not control, part of what we call repeat noise may be drift in the service rather than stochasticity of a fixed model, and the earlier study's backbones were withdrawn before this one began. An open-weights replication is the second thing we would add.

How much floor there is. The earlier data give six repeat pairs at $n{=}120$; the new instruments give sixteen pairs on `gpt-5.6-sol` and three each on `gpt-5.6-luna` and `gpt-6-sol` at $n{=}500$; the three interior temperatures, the raw format, the re-ranker and LoCoMo were run once each. Neither is large, and Appendix F makes the cost explicit: two runs per arm narrow but do not rule out the reported size, and their execution resample rests on two values.

8 CONCLUSION

We set out to test whether relevance is the right objective for selecting injected procedural priors, with a comparison that holds everything but the selection rule fixed. The question cannot be asked at the precision the field has been asking it. Two executions of the identical configuration differ by one to two points of judge accuracy on LongMemEval, and the effects reported in this line of work, the earlier study's included, are of that size. Against that floor, at $n{=}500$ with repeated executions on three current backbones, no arm shows the reported effect. Where the repeats suffice its size is excluded; on the other two backbones two runs per arm cannot tell inertness from noise; single runs of every temperature between the two rules, the raw prior text, an LLM re-ranker and a second benchmark land inside the same floor. The selector collapses onto a few skills, as predicted, and it does not matter. A claim that priors of this kind help or hurt needs to clear a floor that must itself be measured. An identified comparison is necessary but not sufficient: a single run of ours would have been read confidently in either direction. The floor, not the design, is what told us how little there was to interpret.

USE OF AI ASSISTANCE

In accordance with the ICLR 2027 policy on AI assistance, we disclose the following. AI systems (Claude, Anthropic) were used for research ideation and execution: the reframing of the study around the relevance-versus-random comparison was proposed by an AI assistant during re-analysis of the earlier study's item-level outputs, and the same assistant wrote and ran the analysis scripts, the run scripts and the selector-temperature and raw-format patches under the authors' direction. AI assistance was used to draft sections of this paper and to polish the writing; all claims, numbers and their interpretation were checked by the authors against the run artefacts. AI assistance was used for retrieval and discovery of related work; every cited work was verified by the authors against its primary source. No AI assistance was used to generate synthetic data or to prove mathematical claims. All experimental artefacts, including per-question outputs, judge labels and selector traces, are included in the supplementary material so that every number can be recomputed without any AI system.

REFERENCES

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. 2026. AutoSkill: Experience-Driven Lifelong Learning via Skill Self-Evolution. arXiv preprint arXiv:2603.01145.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025. Rethinking Memory in LLM based Agents: Representations, Operations, and Emerging Topics. arXiv preprint arXiv:2505.00675.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2025. Memp: Exploring Agent Procedural Memory. arXiv preprint arXiv:2508.06433.

Shicheng Ye, Chao Yu, Kaiqiang Ke, Chengdong Xu, and Yinqi Wei. 2025. H2R: Hierarchical Hindsight Reflection for Multi-Task LLM Agents. arXiv preprint arXiv:2509.12810.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. arXiv preprint arXiv:2507.05257.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI Agent. arXiv preprint arXiv:2506.06326.

Zhiyu Li, Chenyang Xi, Chunyu Li, et al. 2025. MemOS: A Memory OS for AI System. arXiv preprint arXiv:2507.03724.

Jiafeng Liang, Hao Li, Chang Li, et al. 2025. AI Meets Brain: Memory Systems from Cognitive Neuroscience to Autonomous Agents. arXiv preprint arXiv:2512.23343.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026. MemSkill: Learning and Evolving Memory Skills for Self-Evolving Agents. arXiv preprint arXiv:2602.02474.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating Very Long-Term Conversational Memory of LLM Agents. arXiv preprint arXiv:2402.17753.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. 2026. OpenClaw-RL: Train Any Agent Simply by Talking. arXiv preprint arXiv:2603.10165.

Simon Yu, Gang Li, Weiyan Shi, and Peng Qi. 2025. PolySkill: Learning Generalizable Skills Through Polymorphic Abstraction. arXiv preprint arXiv:2510.15863.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv preprint arXiv:2501.13956.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. SimpleMem: Efficient Lifelong Memory for LLM Agents. arXiv preprint arXiv:2601.02553.

Xiangyi Li, Wenbo Chen, Yimin Liu, et al. 2026. SkillsBench: Benchmarking How Well Agent Skills Work Across Diverse Tasks. arXiv preprint arXiv:2602.12670.

Zhen Tan et al. 2025. In Prospect and Retrospect: Reflective Memory Management for Long-term Personalized Dialogue Agents. arXiv preprint arXiv:2503.08026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. arXiv preprint arXiv:2410.10813.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration. arXiv preprint arXiv:2402.00367.

Cheng Yang, Xuemeng Yang, Licheng Wen, et al. 2025. Learning on the Job: An Experience-Driven Self-Evolving Agent for Long-Horizon Tasks. arXiv preprint arXiv:2510.08002.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv preprint arXiv:2504.19413.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for RAG systems. In *SIGIR*, 2024. arXiv:2401.14887.

Michał Mazuryk, Fleur Dolmans, Louis Gehringer, Ina Klaric, Jia-Huei Ju, and Mohammad Aliannejadi. The powerless noise: How experimental settings shape the reported power of noise. In *SIGIR*, 2026. arXiv:2607.03615.

Kuan Wang. MemDelta: Controlled baselines and hidden confounds in agent memory evaluation. arXiv:2606.29914, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. arXiv:2502.12110, 2025.

A Experimental protocol

Table 3 lists every setting shared across arms. Nothing in it is tuned in this paper; the values are inherited from the earlier study so that the only deliberate change is the comparison itself.

B The bank

The 18 skills are software and office procedures: document co-authoring, spreadsheet and slide manipulation, PDF handling, web-app testing, front-end and canvas design, brand guidelines, algorithmic art, internal communications, skill authoring. None concerns a person's life, preferences or history. This is what makes the bank a clean instrument for the question we ask—a ranking over it has nothing relevant to find—and what makes it unsuitable for asking whether *relevant* priors would help.

C Prior representations

Adapted (default). Each skill's `SKILL.md` was passed once through a fixed adapter prompt to `gpt-5.2-low` and stored as a four-field JSON record. The injected line for a prior is

```
- <skill_id>:  trigger=...| retrieval=...| update=...|
write=...| tags=...
```

truncated to $\lfloor B/k \rfloor = 128$ tokens. The selector matches the question against this adapted text.

Item	Setting	Note
Memory system	SimpleMem, retrieve-only intervention	write side untouched; reflection rounds off (Liu et al., 2026)
Write-side store	`raw-turn`, built once, shared by all arms	identical store for every run
Store embedding	`all-MiniLM-L6-v2` (local)	also the selector's embedding; never varied within a comparison (Wang, 2026)
Retrieval depth	semantic 10, keyword 3, structured 3	SimpleMem defaults
Prior bank	18 skills, `anthropics/skills` at commit `b0cbd3df`	pinned; see Table 4
Adapter	one-shot, `gpt-5.2-low`, frozen JSON	priors are fixed artefacts, not regenerated per backbone
Injection	k=3 priors, B=384 tokens, $\lfloor B/k \rfloor$ per prior	same formatter and budget in every arm
Selector backends	TF–IDF; MiniLM embedding; LLM re-rank of top-10	RANDOM: uniform k-subset, seeded per question
Backbones	`gpt-5.6-sol` (main); `gpt-5.6-luna`, `gpt-6-sol` (replications)	temperature 0; served through a gateway
Judge	LongMemEval yes/no protocol, judge = backbone, `max_tokens`=10	one judge call per question
Questions	LongMemEval-S, all 500; LoCoMo, 200 (seed 42)	LoCoMo scored by category-aware F1, no judge
Repeats	three runs per arm, differing only in run identifier; RANDOM also in seed	the within-arm floor

Table 3: Shared protocol. The earlier study's backbones (`gpt-5.4-high`, `gpt-5.4`) and judge (`gpt-5.2-low`) were withdrawn from the gateway between that study and this one; its numbers are re-read from its own artefacts and never mixed with new ones.

Bank	N	Mean tokens	Median	Min–Max
Curated public skills	18	2354	2031	28–7322

Table 4: Source-skill lengths. A skill is on average six times the whole prior budget and eighteen times the per-prior share of it.

Raw (uncompressed control). Selection is identical—the same selector over the same adapted text—but the injected line carries the original `SKILL.md` instead of the four fields, truncated to the same 128 tokens. Because source skills average 2,354 tokens, "raw" means the head of the file: the frontmatter (name, description) and the opening of the body. The two representations therefore differ in *format* and in *which part* of the skill survives the budget, not in the budget. For 14 of the 18 skills the 128-token head reaches past the frontmatter into the body; for the four with the longest frontmatter (`docx`, `xlsx`, `pptx`, `pdf`, 116–222 frontmatter tokens) it does not, and the planner sees name and description only. If the adapted arm's deficit against RANDOM disappears under the raw format, the deficit was a property of the adapter's output; if it persists, it was not.

Generic guideline. The earlier study's GENERIC arm injects one fixed, hand-written retrieval guideline truncated to B tokens, with no selection. We report it where the earlier data are re-read but do not run it anew; it is not part of the identified contrast.

D THE SELECTOR TEMPERATURE

With $s_p = \mathrm{sim}(q, p)$ over all 18 priors, draw k without replacement from $\mathrm{softmax}(s/\tau)$ via Gumbel-top-k: $\mathrm{key}_p = s_p/\tau + g_p$, $g_p \sim \mathrm{Gumbel}(0, 1)$, take the k largest keys. The Gumbel stream is seeded

by the question id and the run seed, so a run is reproducible and repeats are independent. τ=0 is implemented as plain arg-max and is identical to the RELEVANCE arm; τ=10^6 is indistinguishable from RANDOM (top-1 share of draws 0.22 against uniform 0.17 on a 200-seed offline probe). Interior points $\tau \in \{0.02, 0.05, 0.2\}$ give top-1 shares 0.90, 0.47, 0.23. Similarity scores on this bank have median 0.067, so τ is on the scale of the score spread. Everything downstream of the selected set is unchanged; τ is recorded in the per-question trace and in the run's metrics.

τ	Runs	Judge acc	Δ vs τ=0 (pt)	Disc.	p
0 (RELEVANCE)	3	0.748	—	—	—
0.02	1	0.748	+0.20	25	1.000
0.05	1	0.752	+0.60	27	0.701
0.2	1	0.754	+0.80	22	0.523
∞ (RANDOM)	5	0.749	+0.60	29	0.711

Table 5: Judge accuracy along the selector temperature (Figure 4c), `gpt-5.6-sol`, n=500.

E READING THE FLOOR

Arm	Discordant of 120	Paired Δ (pt)
BASE	4	+0.00
GENERIC	6	+1.67
RANDOM	7	−0.83
RELEVANCE	3	+0.83
NOISY	6	+1.67
MEMSKILL	3	−0.83

Table 6: Two executions of the identical configuration (gpt-5.4-high, TF–IDF selector), ten hours apart, on their 120 shared LongMemEval questions. The largest paired difference between repeats is 1.67 points; the study's reported effects on the same questions are 1.67 to 3.33 points. From `scripts/paired_contrasts.py`.

Arm	Pair	Discordant of 500	Paired Δ (pt)	p
BASE	r1–r2	34	+0.00	1.00
BASE	r1–r3	27	−1.00	0.44
BASE	r2–r3	39	−1.00	0.52
RANDOM	s1–s2	26	−0.40	0.85
RANDOM	s1–s3	32	+0.00	1.00
RANDOM	s1–s4	37	−0.60	0.74
RANDOM	s1–s5	27	−0.60	0.70
RANDOM	s2–s3	22	+0.40	0.83
RANDOM	s2–s4	23	−0.20	1.00
RANDOM	s2–s5	17	−0.20	1.00
RANDOM	s3–s4	23	−0.60	0.68
RANDOM	s3–s5	25	−0.60	0.69
RANDOM	s4–s5	18	+0.00	1.00
RELEVANCE	r1–r2	27	+0.60	0.70
RELEVANCE	r1–r3	25	−0.60	0.69
RELEVANCE	r2–r3	26	−1.20	0.33

Table 7: Within-arm repeat differences at n=500. The maximum $|\Delta|$ in this table is the floor against which Section 6 reads every effect.

For each arm a with runs $r_1, \ldots, r_m$ and for each pair (r_i, r_j), the within-arm difference is the paired accuracy difference over the 500 questions, computed exactly as the between-arm differences are: $\Delta = \big(\#\{q : r_j \text{ right}, r_i \text{ wrong}\} - \#\{q : r_i \text{ right}, r_j \text{ wrong}\}\big)/500$. The floor is $\max_{a,i<j} |\Delta|$. A run enters this computation only if its `metrics.json` matches the reference configuration on every field other than run identity and outcome; the analysis script drops and names any pair that

does not. Every run is also checked for blank or failed hypotheses, which is how a failed model call is recorded, before its labels are used; every run on `gpt-5.6-sol` and `gpt-6-sol` has zero, and the four `gpt-5.6-luna` runs with one to four are kept, scored as wrong, and re-read without the seven affected questions in Section 6.5.

F READING AGAINST A SMALLEST EFFECT OF INTEREST

The within-arm floor is a maximum over few repeat pairs and is used as a scale. The interval the inference rests on comes from a bootstrap (2,000 resamples, `scripts/joint_bootstrap.py`) that resamples the 500 questions with replacement and, within each arm, the executions with replacement, so that execution-to-execution scatter enters the interval; the estimator is the mean over all cross-arm run pairs of the paired accuracy difference. The smallest effect of interest is 2.67 points, the effect the earlier study reported under TF–IDF and the smaller of the two relevance-versus-random readings on its outputs; a size is called excluded only when the interval excludes both $+2.67$ and -2.67. Table 8 gives every contrast. On `gpt-5.6-sol` all three are excluded. With two runs per arm on the other two backbones the intervals are 2.4 to 3.6 points wide on each side, and the reported size survives. With two runs per arm the execution resample takes only four configurations, so the execution term of those intervals is driven by the two observed runs; they are reported as bounds, not as precise widths. Removing the seven `gpt-5.6-luna` questions with a failed answer moves RANDOM→RELEVANCE there from $+1.00$ to $+0.61$ ($[-1.83, +3.14]$).

Backbone	Contrast	Runs	Mean (pt)	95% joint CI	Bound	Excludes ± 2.67
`gpt-5.6-sol`	RANDOM→RELEVANCE	5×3	-0.08	$[-1.65, +1.56]$	1.65	yes
`gpt-5.6-sol`	BASE→RELEVANCE	3×3	-0.47	$[-2.07, +1.20]$	2.07	yes
`gpt-5.6-sol`	BASE→RANDOM	3×5	-0.39	$[-1.84, +1.04]$	1.84	yes
`gpt-5.6-luna`	RANDOM→RELEVANCE	2×2	$+1.00$	$[-1.40, +3.30]$	3.30	no
`gpt-5.6-luna`	BASE→RELEVANCE	2×2	$+0.70$	$[-1.70, +3.20]$	3.20	no
`gpt-5.6-luna`	BASE→RANDOM	2×2	-0.30	$[-2.40, +1.80]$	2.40	yes
`gpt-6-sol`	RANDOM→RELEVANCE	2×2	-1.00	$[-3.60, +1.50]$	3.60	no
`gpt-6-sol`	BASE→RELEVANCE	2×2	-0.20	$[-3.10, +3.00]$	3.10	no
`gpt-6-sol`	BASE→RANDOM	2×2	$+0.80$	$[-1.80, +3.50]$	3.50	no

Table 8: Joint question-and-execution bootstrap intervals for every cross-arm contrast; "Bound" is the larger absolute end of the interval, the size within which the effect is confined at 95%.

Arm	Pair	Discordant of 500	Paired Δ (pt)	p
BASE	r1–r2	33	-0.20	1.00
RANDOM	s1–s2	36	-0.40	0.87
RELEVANCE	r1–r2	34	$+2.00$	0.12

Table 9: Within-arm repeat differences on `gpt-5.6-luna`.

Arm	Runs	Mean acc	Range
BASE	3	0.753	0.746–0.756
RELEVANCE	3	0.748	0.742–0.754
RANDOM	5	0.749	0.746–0.752

Contrast	Run pairs	Mean Δ (pt)	Range	W/L pooled	p	Reading (floor 1.20)
BASE→RELEVANCE	9	-0.47	-1.40 to $+0.80$	111/132	0.199	inside floor
BASE→RANDOM	15	-0.39	-1.00 to $+0.60$	190/219	0.166	inside floor
RANDOM→RELEVANCE	15	-0.08	-1.00 to $+0.80$	206/212	0.807	inside floor

Table 10: `gpt-5.6-sol`, n=500. Top: per-arm judge accuracy over independent runs. Bottom: cross-arm paired differences over all run pairs, pooled discordant counts, exact McNemar p on the pooled counts (run pairs share runs and are not independent; the pooled p is reported as a summary, not a test), and the prospective reading against the within-arm floor of Table 7.

Arm	Runs	Mean acc	Range
BASE	2	0.727	0.726–0.728
RELEVANCE	2	0.734	0.724–0.744
RANDOM	2	0.724	0.722–0.726

Contrast	Run pairs	Mean Δ (pt)	Range	W/L pooled	p	Reading (floor 2.00)
BASE→RELEVANCE	4	+0.70	−0.40 to +1.80	78/64	0.275	inside floor
BASE→RANDOM	4	−0.30	−0.60 to +0.00	68/74	0.675	inside floor
RANDOM→RELEVANCE	4	+1.00	−0.20 to +2.20	71/51	0.085	inside floor

Table 11: `gpt-5.6-luna`, n=500, two runs per arm. Same layout as Table 10; the floor is this backbone's own within-arm maximum (2.00 points, Table 9). Four of the six runs carry one to four failed answers from transient gateway errors (Appendix G), scored as wrong; without the seven affected questions RANDOM→RELEVANCE is +0.61 and the floor 2.23.

Arm	Runs	Mean acc	Range
BASE	2	0.729	0.718–0.740
RELEVANCE	2	0.727	0.718–0.736
RANDOM	2	0.737	0.732–0.742

Contrast	Run pairs	Mean Δ (pt)	Range	W/L pooled	p	Reading (floor 2.20)
BASE→RELEVANCE	4	−0.20	−2.20 to +1.80	78/82	0.813	inside floor
BASE→RANDOM	4	+0.80	−0.80 to +2.40	85/69	0.227	inside floor
RANDOM→RELEVANCE	4	−1.00	−2.40 to +0.40	67/87	0.125	inside floor

Table 12: `gpt-6-sol`, n=500, two runs per arm; floor 2.20 points (Table 13).

G RUN MANIFEST

Runs excluded. On 2026-09-21 the API account behind the gateway ran out of balance during the sol lane. The memory system records a failed generation as the literal hypothesis "Failed to generate answer", which the judge and the F1 scorer then mark wrong; a blank-hypothesis check does not catch it. Every run answered after that point was quarantined and re-run once the account was restored; each rerun keeps the original run identifier, so the same string appears failed in Table 15 and clean in the manifest. A fourth backbone, `gpt-6-luna`, was attempted and abandoned: the gateway rate-limits it per model so severely that its RELEVANCE run returned 43 failed answers of 500 and its judge could not complete; only its clean BASE run exists and nothing from it is reported. Table 15 lists the excluded runs so that the exclusion is auditable.

H USE OF THE EARLIER STUDY'S ARTEFACTS

The earlier study's per-question predictions, judge labels and selector traces for its March (TF–IDF, `gpt-5.4-high`) and May (embedding, `gpt-5.4`) runs are included in the supplementary material, together with the two scripts that produce every number in Section 6.1, Section 4 and Table 6 from them without any model call.

Arm	Pair	Discordant of 500	Paired Δ (pt)	p
BASE	r1–r2	41	+2.20	0.12
RANDOM	s1–s2	31	+1.00	0.47
RELEVANCE	r1–r2	39	−1.80	0.20

Table 13: Within-arm repeat differences on `gpt-6-sol`.

Backbone	Arm	Run	n	Blank	Judge acc	Mean tokens
5.6-sol	base	sol_base_r1	500	0	0.756	4975
5.6-sol	base	sol_base_r2	500	0	0.756	4954
5.6-sol	base	sol_base_r3	500	0	0.746	4969
5.6-sol	random	sol_random_s1	500	0	0.752	5764
5.6-sol	random	sol_random_s2	500	0	0.748	5728
5.6-sol	random	sol_random_s3	500	0	0.752	5714
5.6-sol	random	sol_random_s4	500	0	0.746	5761
5.6-sol	random	sol_random_s5	500	0	0.746	5738
5.6-sol	raw-random	sol_raw_random_s1	500	0	0.738	5701
5.6-sol	raw-relevance	sol_raw_curated_r1	500	0	0.748	5731
5.6-sol	relevance	sol_curated_r1	500	0	0.748	5728
5.6-sol	relevance	sol_curated_r2	500	0	0.754	5736
5.6-sol	relevance	sol_curated_r3	500	0	0.742	5756
5.6-sol	rerank	sol_rerank_curated_r1	500	0	0.748	5737
5.6-sol	tau=0.02	sol_dial_tau0.02	500	0	0.748	5765
5.6-sol	tau=0.05	sol_dial_tau0.05	500	0	0.752	5770
5.6-sol	tau=0.2	sol_dial_tau0.2	500	0	0.754	5750
5.6-luna	base	floor_base_r1	500	1	0.728	4755
5.6-luna	base	floor_base_r2	500	0	0.726	4739
5.6-luna	random	head_random_s1	500	4	0.726	5446
5.6-luna	random	head_random_s2	500	1	0.722	5617
5.6-luna	relevance	head_curated_r1	500	0	0.724	5624
5.6-luna	relevance	head_curated_r2	500	1	0.744	5498
6-sol	base	gpt6sol_base_r1	500	0	0.718	4780
6-sol	base	gpt6sol_base_r2	500	0	0.740	4860
6-sol	random	gpt6sol_random_s1	500	0	0.732	5443
6-sol	random	gpt6sol_random_s2	500	0	0.742	5451
6-sol	relevance	gpt6sol_curated_r1	500	0	0.736	5617
6-sol	relevance	gpt6sol_curated_r2	500	0	0.718	5587

Table 14: Every judged run behind the numbers in this paper. A blank hypothesis is empty text or the literal failed-generation string; both record a failed model call. Every `gpt-5.6-sol` and `gpt-6-sol` run has zero; the four `gpt-5.6-luna` runs with one to four are kept, scored as wrong, and re-read without the affected questions in Section 6.5.

Run	n	Failed answers	Reason
LoCoMo__gpt-5.6-sol/base/sol_locomo200_r1	200	200	gateway balance exhausted mid-run
LoCoMo__gpt-5.6-sol/curated/sol_locomo200_r1	200	200	gateway balance exhausted mid-run
LoCoMo__gpt-5.6-sol/random/sol_locomo200_r1	200	200	gateway balance exhausted mid-run
dial__gpt-5.6-sol/curated/sol_dial_tau0.02	500	500	gateway balance exhausted mid-run
dial__gpt-5.6-sol/curated/sol_dial_tau0.05	500	500	gateway balance exhausted mid-run
dial__gpt-5.6-sol/curated/sol_dial_tau0.2	500	500	gateway balance exhausted mid-run
random__sol_random_s4	500	29	gateway balance exhausted mid-run
random__sol_random_s5	500	458	gateway balance exhausted mid-run

Table 15: Runs excluded from every number in this paper, with the count of failed answers that triggered the exclusion.