PUREMEM: LEARNER-RELATIVE MEMORY CURATION FOR RELIABLE RETRIEVAL-AUGMENTED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented language models commonly assume that relevant and correct context is useful. This assumption can fail when retrieved memory is redundant, difficult for the target model to exploit, or induces an incompatible reasoning strategy. We introduce PUREMEM, a learner-relative memory framework that asks not only what is relevant, but what a frozen language model can reliably use. PureMem constructs a compact reasoning memory bank from examples in the model's empirically estimated zone of proximal development (ZPD), and retrieves memory through three stages: candidate recall, structured procedure matching, and LLM-based applicability verification. A conservative injection policy falls back to the base model when no candidate is judged applicable. On BBEH, PureMem improves accuracy from 72.75% to 74.78% while retaining only 26.4% of the full memory bank. Counterfactual placebo interventions show that the verifier separates relatively beneficial memory from harmful memory: injected memory improves accepted cases but significantly degrades rejected cases. On RealMemBench, the three-stage retrieval harness improves NDCG@10 by 0.097 over the strongest executed embedding baseline and also improves downstream answer quality. These results distinguish a model-specific memory-curation procedure from a reusable retrieval architecture, while showing that injection policies must be adapted to the target memory domain. Code and data are provided as anonymised supplementary material.

1 INTRODUCTION

Large language models (LLMs) can be augmented with external information through retrieval-augmented generation (RAG) (Guu et al., 2020; Lewis et al., 2020), dense passage retrieval (Karpukhin et al., 2020), and in-context learning (ICL) (Brown et al., 2020). Subsequent systems have extended this paradigm to large-scale external memory and retrieval-augmented few-shot learning (Borgeaud et al., 2022; Izacard et al., 2023). These paradigms implicitly rely on the assumption that more relevant context will lead to better generation.

However, textual relevance does not necessarily imply usefulness to a particular model. Prior work has shown that language models may fail to effectively use relevant information when it is buried in long contexts, mixed with distractors, or presented in an incompatible format (Liu et al., 2024; Shi et al., 2023). Even a semantically relevant precedent may be redundant when it supplies information the model already has, too difficult to exploit reliably when it exceeds the model's current reasoning capability, or misleading when it instantiates an incompatible reasoning procedure (Min et al., 2022; Liu et al., 2022; Margatina et al., 2023); it also consumes context budget, and can turn an otherwise correct prediction into an incorrect one. These failure modes persist even when demonstrations are selected by careful relevance-based retrieval (Srivastava et al., 2024; Cui & Sachan, 2025). The central problem is therefore not only *which memory is relevant to a query*, but also:

> *Which examples are useful sources of memory for this particular frozen language model, and which retrieved memories should be injected for the current query?*

We address this problem with PUREMEM, a learner-relative memory framework for retrieval-augmented reasoning. PUREMEM views memory construction as a form of learner-specific curriculum design rather than indiscriminate corpus collection, connecting external-memory construction to

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.

learner-aware curriculum and self-paced learning (Bengio et al., 2009; Kumar et al., 2010). In particular, it uses the Zone of Proximal Development (ZPD) (Vygotsky & Cole, 1978) to identify promising *source examples*. Given repeated attempts by a frozen student model on candidate training problems, PUREMEM estimates the model's pass rate and selects examples with intermediate difficulty: examples that the model does not reliably solve alone, but that are not entirely beyond its effective learning range. The operational hypothesis is that very easy examples carry little additional information while far-out-of-reach examples cannot be exploited reliably; we test it empirically rather than assuming that intermediate difficulty is universally optimal. The selected ZPD examples are then passed to a teacher model, which generates answer-verified solution traces. These traces are decomposed into concrete precedents and mechanism-level procedural schemas, forming a structured external memory bank.

This construction procedure makes an important distinction between *source-level curation* and *inference-time memory selection*, two decisions that are often implemented at different stages in retrieval-augmented systems (Jiang et al., 2023; Asai et al., 2024; Yan et al., 2024). ZPD pass rate is the primary signal used to select memory sources; the generated chunks are not individually admitted through a chunk-level counterfactual utility test. Instead, PUREMEM controls memory use at inference time through an applicability gate. The gate evaluates whether a retrieved precedent or procedural schema offers a compatible approach for the current query and injects memory only when at least one candidate passes verification. When no candidate is accepted, PUREMEM falls back to the same prompt used by the no-memory baseline.

The inference-time retrieval process is organized as a three-stage decision funnel. First, *Candidate Recall* retrieves potentially relevant episodes, precedents, or sessions with high recall. Second, *Structured Matching* uses procedural structure and available metadata to identify candidates that share a reasoning mechanism or memory relation with the query. Third, *Utility Verification* uses an LLM-based ranker to assess query–memory applicability and rerank the candidates. Separating broad recall from procedural matching and from the final injection decision lets the retrieval interface stay consistent across memory domains while leaving the memory objects and injection policy task-dependent.

PUREMEM consequently separates three questions that retrieval-augmented systems often conflate: *what should the LLM remember?*—answered by ZPD-based source curation over learner-relative examples; *what memory is potentially relevant?*—answered by the hierarchical retriever; and *what memory should be used now?*—answered by the applicability gate, which injects only compatible candidates instead of transferring memory indiscriminately.

We evaluate these components on two benchmarks that test complementary claims. BBEH (Kazemi et al., 2025) tests learner-relative curation and task-specific injection control for hard textual reasoning: besides comparing ZPD-selected memory against full and size-matched control banks, we use paired counterfactual placebo interventions to ask whether the inference-time gate separates helpful injections from ones that would cause negative transfer. RealMemBench (Bian et al., 2026) tests the retrieval architecture alone as a harness for long-term conversational memory, where temporally preceding sessions form the corpus and neither ZPD probing nor teacher-trace construction applies.

Our contributions are: (i) PUREMEM, a learner-relative memory framework that builds external reasoning memory from source examples in a frozen model's empirically estimated ZPD; (ii) a structured construction procedure that turns teacher-generated, answer-verified traces into concrete precedents and mechanism-level procedural schemas; (iii) a three-stage retrieval harness combining candidate recall, procedure-aware structured matching, and LLM-based applicability verification for selective injection; and (iv) paired evaluations that separate memory curation from retrieval, using size-matched controls and counterfactual placebo analysis on BBEH and a transfer test on REALMEM.

2 RELATED WORK

Retrieval-augmented generation and external memory. Retrieval-augmented generation supplies language models with text from an external corpus (Guu et al., 2020; Lewis et al., 2020), and later systems scale the corpus, the retriever, or the memory available to a long-context model (Khandelwal et al., 2020; Borgeaud et al., 2022; Izacard et al., 2023; Wu et al., 2022; Wang et al., 2023b). These

systems optimize corpus coverage, retrieval recall, and the relevance of the supplied context while treating the external corpus as fixed. PUREMEM instead makes the composition of the bank model-dependent, asking whether an item is usable by a particular frozen model rather than merely relevant to the query.

Demonstration selection and learner-aware curation. In-context demonstration selection has been studied through semantic relevance, diversity, ordering, and retrieval quality (Liu et al., 2022; Rubin et al., 2022; Lu et al., 2022; Min et al., 2022), and through adaptive or active strategies driven by model behavior (Margatina et al., 2023; Su et al., 2022); demonstrations are also known to act on the model beyond supplying task labels (Zhao et al., 2021), which is what motivates our distinction between textual relevance and model-specific usability. These methods select examples for an individual prompt at inference time. PUREMEM differs in granularity and purpose: it estimates a frozen student's empirical pass rate on candidate sources, keeps the intermediate-difficulty ones, and builds reusable memory before any test query is seen. The criterion follows the Zone of Proximal Development (Vygotsky & Cole, 1978) and is related to curriculum and self-paced learning (Bengio et al., 2009; Kumar et al., 2010), but schedules no parameter updates—learner-relative difficulty serves purely as a curation signal.

Long-term memory for language-model agents. Agent memory systems maintain episodic, semantic, or structured summaries of past interaction and retrieve them in later decisions (Park et al., 2023; Shinn et al., 2023; Wang et al., 2023a; Packer et al., 2023; Zhong et al., 2024), with recent benchmarks and retrievers stressing temporal reasoning and long-term recall (Maharana et al., 2024; Gutiérrez et al., 2024). PUREMEM shares structured memory objects and multi-stage retrieval but carries a different payload—curated reasoning precedents and procedural schemas rather than accumulated history—and we use REALMEM to test only whether the retrieval harness transfers to that setting, assuming neither that ZPD construction nor that the conservative injection policy does.

3 METHOD

3.1 OVERVIEW AND PROBLEM SETUP

Let $\mathcal{D}_{\text{src}} = \{(x_i, y_i)\}_{i=1}^{N}$ be an external pool of solved source problems, and let M denote a frozen student language model. Our goal is to construct an external memory bank $\mathcal{B}$ that improves reasoning on unseen queries without updating M. PUREMEM separates this goal into two decisions: (i) selecting source examples from which memory should be constructed, and (ii) deciding whether retrieved memory should be injected for a particular test query. Figure 1 gives an overview.

The first decision is made during memory construction. Rather than treating all source examples as equally useful, PUREMEM estimates their learner-relative difficulty for M and retains examples in the model's empirically estimated Zone of Proximal Development (ZPD). The second decision is made at inference time. Given a query, PUREMEM retrieves candidate memories through three stages and injects them only when an applicability verifier accepts at least one candidate. Thus, ZPD is a source-curation signal, whereas the verifier is an inference-time negative-transfer control mechanism.

3.2 ZPD-BASED SOURCE CURATION

For each source problem x_i, we sample the frozen student model m times under the no-memory prompt and compute its empirical pass rate:

$$\hat{p}_i = \frac{1}{m} \sum_{r=1}^{m} \mathbb{I}\Big[M^{(r)}(x_i) = y_i\Big]. \tag{1}$$

Here, $M^{(r)}(x_i)$ is the prediction from the r-th independent attempt, and correctness is determined by the task's programmatic evaluator. In the implementation the denominator is the number of attempts that actually returned a scorable completion: an attempt lost to an infrastructure failure is excluded rather than counted as incorrect, since a failed request carries no evidence about whether the student can solve the item.

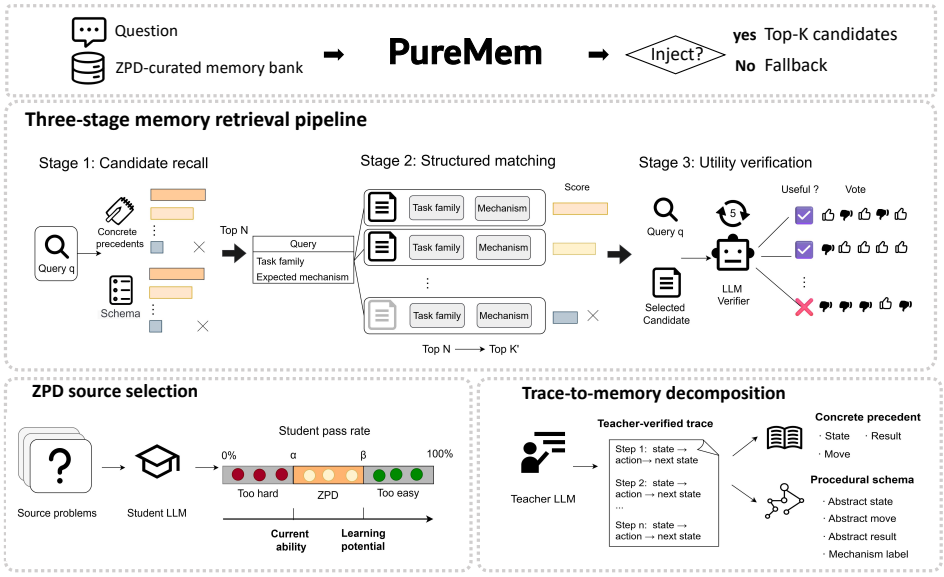


Figure 1: **Overview of PUREMEM**, which separates learner-relative memory construction from inference-time memory selection. **Top:** given a query and the ZPD-curated bank, PUREMEM decides whether to inject memory at all, falling back to the prompt-matched no-memory solver when no candidate is judged applicable. **Middle:** the three-stage pipeline recalls a broad candidate set from both the precedent and the schema view, reranks it by adding task-family and mechanism-label agreement to the embedding score, and verifies utility with an LLM gate sampled N times and aggregated by majority vote; only the top-K candidates clearing the vote are injected. **Bottom left, (a) ZPD source selection:** the frozen student's pass rate is estimated from repeated attempts, and only sources of intermediate difficulty are retained—those it already solves reliably are redundant, those it essentially never solves it cannot exploit. **Bottom right, (b) Trace-to-memory decomposition:** a teacher LLM produces answer-verified traces for retained sources, and each reasoning step yields a concrete precedent—prior state, concrete move, resulting state—paired one-to-one with a procedural schema that abstracts the same triple and labels its mechanism.

We retain a source example when its pass rate lies within an intermediate range:

$$\mathcal{D}_{\mathrm{ZPD}} = \{(x_i, y_i) \in \mathcal{D}_{\mathrm{src}} \mid \alpha \leq \hat{p}_i \leq \beta\}\,, \tag{2}$$

where α and β define the ZPD interval. We use the following operational hypothesis: high-pass-rate examples are more likely to be redundant, whereas very-low-pass-rate examples may be difficult for the student to exploit reliably. The validity of this hypothesis is evaluated through the size-matched control experiments in Sec. 5.1. The retained examples instead represent problems for which external procedural support is potentially useful.

Importantly, Eq. (2) selects *source examples*, not individual memory chunks. PUREMEM does not claim that every chunk subsequently derived from a ZPD source has been causally verified at admission time. The counterfactual analyses in our experiments instead evaluate whether the inference-time verifier distinguishes relatively helpful injections from harmful ones.

3.3 MEMORY CONSTRUCTION

For each ZPD-selected source example, a teacher model produces a chain-of-thought solution trace (Wei et al., 2022) conditioned on the problem and its verified answer. We retain only answer-consistent traces, a rejection-sampling criterion analogous to the answer-verified rationale filtering used in self-taught reasoning (Zelikman et al., 2022) and rationale distillation (Hsieh et al., 2023), except that the accepted traces populate an external memory bank rather than a fine-tuning set.

Each retained trace is split into its individual reasoning steps, and every step yields two complementary views. A *concrete precedent* $c_{i,j} = (s_{i,j}, a_{i,j}, s'_{i,j})$ records the state at the start of the step, the

concrete reasoning move, and the state that move establishes, keeping the specific quantities and objects that let it be recognized by analogy. The originating problem statement is not carried into the bank, so a precedent is a worked *step* rather than a worked problem—small enough that several fit alongside a BBEH question. A one-to-one linked *procedural schema* mirrors the same triple with instance-specific entities and values removed, and adds a mechanism label from a fixed inventory of reasoning types such as constraint propagation, elimination, state tracking, or string manipulation. Abstraction is partial by design: traversal direction, cardinality, the named operation, and any condition gating the move all survive, because an abstractor instructed to strip everything specific collapses distinct mechanisms into a single content-free shell that Stage III can no longer discriminate among. Writing $\mathcal{C}_i$ and $\mathcal{A}_i$ for the precedents and schemas derived from source i, the reasoning memory bank is

$$\mathcal{B} = \bigcup_{(x_i,y_i)\in\mathcal{D}_{\mathrm{ZPD}}} (\mathcal{C}_i \cup \mathcal{A}_i)\,. \tag{3}$$

Field-level definitions, the design of the label inventory, and the fallback applied when schema abstraction fails validation are given in Sec. A.4.

3.4 Three-Stage Retrieval and Verification

Given a test query q, PureMem progressively narrows the set of memories that may be injected. The three stages separate broad candidate recall, structured matching, and final applicability verification.

Stage I: Candidate Recall. We encode the query using a sentence-embedding model E (Reimers & Gurevych, 2019):

$$\mathbf{z}_q = E(q). \tag{4}$$

For reasoning memory, we retrieve top-N_A procedural schemas and top-N_C concrete precedents using semantic similarity. Each retrieved schema contributes its linked precedent, and the union forms a high-recall candidate pool $\mathcal{P}(q)$. This dual representation allows retrieval to match either surface-level problem similarity or an abstract reasoning procedure.

Stage II: Structured Matching. Semantic similarity alone may retrieve examples with incompatible solution mechanisms. We therefore rerank candidates using their query–memory structural compatibility:

$$s_{\mathrm{match}}(q,c) = \cos(\mathbf{z}_q,\mathbf{z}_c) + \lambda_1\,\mathrm{sim}_{\mathrm{proc}}(q,c) + \lambda_2\,\mathrm{sim}_{\mathrm{meta}}(q,c), \tag{5}$$

where $\mathrm{sim}_{\mathrm{proc}}$ tests whether a candidate's mechanism label lies within the set of mechanisms expected for the query, and $\mathrm{sim}_{\mathrm{meta}}$ tests agreement on the available structured field—the task family on BBEH, and the conversational topic on RealMem. Both terms are indicator functions rather than graded similarities, so Eq. (5) adds a bounded bonus to the embedding score rather than replacing it. Keeping them additive is a deliberate choice: a hard same-task filter would reduce the system to per-task few-shot retrieval, and cross-task mechanism transfer is precisely what we want the design to permit.

Because a test query has no trace of its own, and therefore no mechanism label, the expected mechanism set is not parsed from the query. It is taken from the empirical mechanism distribution of the query's task family among the training precedents, which is available at query time and uses no test-side annotation (Sec. A.5). We retain the top-K' matched candidates for final verification.

Stage III: Utility Verification. For each shortlisted candidate, an LLM-based verifier (Zheng et al., 2023; Sun et al., 2023) receives the query and the candidate memory, including the schema and linked precedent when available. It evaluates whether the candidate provides a compatible and useful approach for solving the current query. Because a single verifier sample is unstable near the decision boundary, we sample the verifier repeatedly and aggregate by majority vote, following the practice of aggregating multiple stochastic samples rather than trusting one (Wang et al., 2023c). Let $r_n(q,c) \in [0,1]$ be the score from the n-th verifier sample. We aggregate N samples as

$$\begin{aligned} V(q,c) &= \sum_{n=1}^{N} \mathbb{I}[r_n(q,c) \geq \tau]\,, \\ \bar{r}(q,c) &= \frac{1}{N}\sum_{n=1}^{N} r_n(q,c). \end{aligned} \tag{6}$$

A candidate is accepted if $V(q, c) \geq v$. Accepted candidates are ranked by $(V(q, c), \bar{r}(q, c))$, and the top-K items form the retrieved set $\mathcal{R}(q)$. If no candidate is accepted, $\mathcal{R}(q) = \emptyset$.

The verifier is deliberately conservative: it does not establish that an accepted item is universally beneficial, nor does it serve as a memory-bank admission test. Rather, it estimates whether a retrieved item is applicable to the current query and prevents indiscriminate injection of potentially harmful memory.

3.5 Selective Memory Injection

The final prediction policy is

$$\hat{y} = \begin{cases} M(\text{Prompt}(q, \mathcal{R}(q))), & \text{if } |\mathcal{R}(q)| > 0, \\ M(\text{Prompt}(q, \varnothing)), & \text{otherwise.} \end{cases} \tag{7}$$

The empty-memory prompt is identical to the no-memory baseline prompt. Therefore, when all candidates are rejected, PureMem falls back to the base model without introducing a prompt-format confound. When memory is injected, the model is instructed to use it as a procedural hint rather than copy instance-specific entities, intermediate states, or answers.

Instantiation across memory domains. On BBEH, $\mathcal{B}$ consists of ZPD-curated reasoning precedents and procedural schemas constructed from an external, disjoint source pool. On RealMem, the benchmark's temporally preceding conversational sessions serve as the memory corpus. In that setting, ZPD probing and teacher-generated reasoning-memory construction are not applied; instead, the same three-stage retrieval interface is instantiated over historical sessions and their available metadata. This distinction allows us to evaluate model-specific memory curation on BBEH separately from the transferability of the retrieval harness on RealMem.

4 Experimental Setup

Protocol. BBEH evaluates ZPD-based source curation and selective injection for hard textual reasoning. The memory bank is built from an external pool of solved reasoning examples disjoint from the evaluation set; evaluation questions, answers, rationales, and test-derived abstractions are used for neither source selection, memory construction, threshold selection, nor prompt development. All BBEH arms are scored on the same 690 shared scorable instances under one frozen base model, task instruction set, decoding configuration, generation budget, extraction procedure, and context budget, and the memory-augmented prompt with an empty memory slot is byte-identical to the no-memory prompt (Sec. A.1); this parity is what makes the comparison a measurement of memory rather than of prompt wording. RealMem instead tests whether the retrieval architecture transfers to a qualitatively different memory domain, where memory consists of temporally preceding conversational sessions and the benchmark's streaming protocol restricts retrieval to sessions available before the current query. That track applies neither ZPD probing, teacher-trace generation, nor schema construction, so it evaluates the transferability of the retrieval harness alone.

Baselines and metrics. On BBEH we compare four memory banks under the shared protocol: **No Memory** (frozen base model, no retrieved context); **Full Memory** (all available source examples and their derived items); **Random Matched** (a random subset matched approximately to the size of the PureMem bank); and **Stratified Matched** (a size-matched subset preserving the source pool's task-family composition). The primary metric is exact-match accuracy under the benchmark's programmatic evaluator, reported alongside source-demonstration count, retained-chunk count, and relative bank size. On RealMem we compare a single-pool embedding retriever (`concrete_only`), a stronger embedding baseline (`simple_embedding`), Stage I–II retrieval without final verification (`stage2_only`), the complete three-stage retriever (`three_stage_rerank`), and the same retriever under the conservative reasoning-oriented gate (`three_stage_gated`), reporting Recall@k and NDCG@k and, on the QA subset, answer quality, memory recall, memory helpfulness, and conflict rate. Metric definitions, prompt-parity, placebo-intervention, and statistical-testing protocols are in Sec. A.

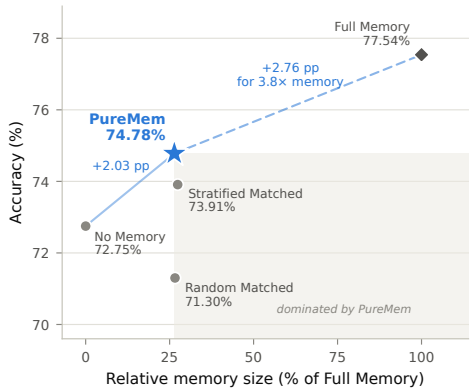


Figure 2: **BBEH accuracy–memory-size trade-off.** PUREMEM improves on the prompt-matched no-memory baseline with 26.4% of the Full Memory bank and beats both size-matched controls; Full Memory is most accurate in absolute terms at far greater capacity.

5 RESULTS

5.1 SAFE AND COMPACT REASONING MEMORY ON BBEH

Figure 2 shows the accuracy–memory-size trade-off on the 690 shared scorable BBEH instances under the unified protocol. Full Memory is most accurate in absolute terms (77.54%), confirming that the complete bank holds useful reasoning support when capacity is unconstrained, but it costs $3{,}616$ chunks. PUREMEM retains 955—26.4% of that bank—while improving over the prompt-matched No Memory baseline from 72.75% to 74.78% ($+2.03$ points), and at a comparable budget it beats Random Matched by 3.48 points and Stratified Matched by 0.87. The improvement is therefore not attributable to compression alone. Since Full Memory remains stronger in absolute accuracy, these results position ZPD curation as a favorable compactness–performance trade-off—a positive aggregate gain over no external memory at roughly a quarter of the capacity—rather than as dominating exhaustive retention.

5.2 WHY PUREMEM WORKS: CURATION AND NEGATIVE-TRANSFER CONTROL

Curation isolates a compact useful source set. Random Matched and Stratified Matched hold the retained chunk count fixed but do not select sources by the frozen student's empirical pass rate, and Fig. 2 shows PUREMEM beating both at a comparable budget. The improvement over No Memory is therefore not explained by compression alone; task-stratified sampling narrows the gap, but learner-relative difficulty carries additional signal.

Counterfactual analysis of the applicability verifier. Figure 3 tests whether Stage III prevents harmful inference-time transfer. Holding the retrieved Top-1 candidate, gate decision, prompt structure, model, and decoding configuration fixed, we compare *Real Top-1*, which inserts the retrieved memory, against a neutral *Placebo* occupying the same slot through the same template but carrying no task-relevant information, over the 611 queries untruncated in both arms (Sec. A.8).

On the 178 accepted queries, inserting the retrieved memory improves accuracy by $+2.81$ points; on the 433 rejected queries, counterfactually inserting the same candidates costs -3.23 points, a paired degradation significant under an exact McNemar test ($p = 0.0094$). The resulting selectivity interaction is $+6.04$ points (95% CI $[+1.82, +10.47]$), significant under a gate-label permutation test ($p = 0.0065$). The two subsets carry unequal evidential weight, however: the Accept-side gain is directionally positive but not individually significant ($p = 0.227$), so what the permutation test certifies is the *contrast*, driven primarily by the reliable Reject-side degradation. The evidence thus supports the gate as a negative-transfer filter rather than a positive-utility oracle.

Role of the conservative injection policy. We ablate Stage III with the ZPD-curated bank and the rest of the inference configuration held fixed (Tab. 1). Under dense retrieval, disabling the gate

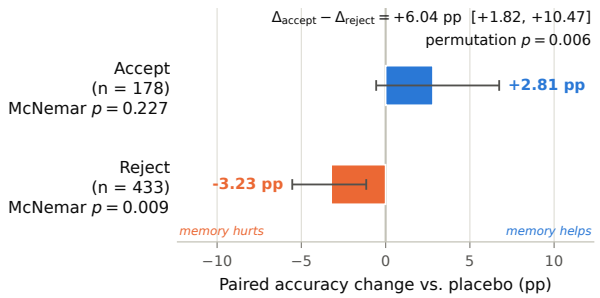


Figure 3: **The applicability verifier separates helpful from harmful memory transfer on BBEH.** Paired accuracy change from inserting the frozen retrieved Top-1 memory rather than a neutral placebo in the same prompt slot, split by the frozen gate decision; error bars are paired bootstrap 95% confidence intervals. Retrieved memory helps on accepted queries but hurts when counterfactually injected on rejected ones. Queries truncated in either arm are excluded (Sec. A.8).

Table 1: Ablation of candidate retrieval and Stage-III applicability verification on BBEH under the unified prompt protocol. All retrieval variants use the compact ZPD-curated memory bank.

Variant	Gate	Accuracy	Injection Rate
No Memory	–	72.75%	0.0%
Dense Retrieval	On	74.78%	36.1%
Dense Retrieval	Off	74.35%	100.0%
Random Retrieval	On	**75.65%**	18.3%
Random Retrieval	Off	73.19%	100.0%

costs 0.43 points and forces injection on every query, so verification contributes modestly while cutting the injection rate from 100.0% to 36.1%. The effect sharpens when candidate quality is deliberately weakened: under random retrieval the gate is worth +2.46 points (73.19% → 75.65%) while injecting for only 18.3% of queries, which is what one expects if its role is to screen a pool containing many incompatible memories. Because random retrieval with the gate slightly exceeds dense retrieval with the gate here, we do not read this ablation as establishing the superiority of dense over random candidate recall; the supported claim is narrower, that Stage III reduces the risk of indiscriminate injection.

5.3 Transfer of the Retrieval Harness to RealMemBench

We now test the retrieval architecture in long-term conversational memory, where the memory objects are historical sessions rather than distilled reasoning precedents. The complete three-stage retriever is strongest on every reported retrieval metric (Tab. 2): against `simple_embedding`, the strongest baseline executed as an independent arm, it improves NDCG@10 from 0.586 to 0.683 and Recall@5 from 0.728 to 0.823. The component pattern follows the intended division of labor, with stronger recall and structured matching improving ranking quality and Stage III reranking contributing the largest gain at the top of the list (+0.062 NDCG@10 over `stage2_only`).

These gains carry downstream. On the QA subset of 1,162 queries, `three_stage_rerank` scores 1.839 against 1.719 for `simple_embedding`, a paired improvement of +0.117 (95% CI [+0.060, +0.176]), with higher memory recall and helpfulness and a lower conflict rate (Tab. 3).

The conservative reject-unless-useful gate, by contrast, does not transfer: `three_stage_gated` is worse than the ungated reranker on both retrieval and QA. This clarifies the boundary of transfer. The retrieval harness is reusable across memory domains, but the injection policy must match the cost structure of the target task: in reasoning an incompatible precedent can distort a solution trajectory, whereas in conversational memory omitting a relevant session is typically worse than admitting a moderately noisy one.

Table 2: Retrieval performance on REALMEM over 1,415 queries from 10 personas.

Method	Recall@5	Recall@20	NDCG@5	NDCG@10	NDCG@20
`concrete_only`	0.661	0.871	0.482	0.528	0.560
`simple_embedding`	0.728	0.924	0.538	0.586	0.618
`stage2_only`	0.772	0.954	0.572	0.621	0.653
`three_stage_rerank`	**0.823**	**0.956**	**0.643**	**0.683**	**0.707**
`three_stage_gated`	0.692	0.700	0.554	0.559	0.559

Table 3: Downstream answer quality on the REALMEM QA subset.

Method	QA Score	Memory Recall	Memory Helpfulness	Conflict Rate
`no_memory`	1.252	–	–	0.290
`simple_embedding`	1.719	0.398	1.297	0.291
`three_stage_rerank`	**1.839**	**0.464**	**1.429**	**0.263**
`three_stage_gated`	1.778	0.363	1.346	0.262

Limitations of this transfer evaluation. Three caveats bound the above. First, `concrete_only` and `stage2_only` are reconstructed offline from the Stage-II ranks recorded by the executed three-stage run rather than run as independent arms, so they attribute the final ranking rather than constituting separate systems; because all arms are scored over the same returned depth of 20, Recall@k for $k \geq 20$ coincides by construction and we therefore claim only top-of-list ranking quality. Second, part of the gated arm's retrieval deficit is mechanical: it returns 2.6 items per query against 19.5 for the other arms, and Recall@k penalizes a list shorter than k however well ordered—though the QA-side drop, which is not normalized by list length, points the same way. Third, Tab. 2 spans all ten personas including the two development personas on which metrics run consistently higher, so it is descriptive rather than held out; Tab. 3 covers the eight test personas only. See Sec. A.7.

6 CONCLUSION

We introduced PUREMEM, a learner-relative framework that distinguishes three questions often conflated in retrieval-augmented systems: what a frozen model should remember, what memory is relevant, and what retrieved memory should be used for a particular query. PUREMEM uses empirical pass rates to curate memory *sources* from the model's Zone of Proximal Development, distills them into concrete precedents and procedural schemas, and retrieves them through Candidate Recall, Structured Matching, and Utility Verification.

On BBEH, PUREMEM improves over the prompt-matched no-memory baseline (74.78% versus 72.75%) with only 26.4% of the full memory bank and beats size-matched random and stratified banks, while counterfactual placebo interventions show the verifier separating subsets with significantly different injection effects. On REALMEM, the harness improves NDCG@10 by 0.097 over the strongest executed embedding baseline and improves answer quality, so the retrieval architecture transfers beyond reasoning precedents.

Boundaries follow. Full Memory is most accurate in absolute terms, making ZPD curation a compactness–performance trade-off rather than a replacement for exhaustive retention, and the band criterion remains an operational hypothesis until the mastered-only and out-of-reach-only controls are run. Because the reject-unless-useful gate does not transfer to conversational memory, injection policies are domain-specific even where the harness is not. Future work should pursue direct utility-based chunk admission on held-out, source-disjoint queries, those two missing controls, adaptive injection policies sensitive to task-specific error costs, and evaluation across models and memory scales. More broadly, external memory should be designed not only around what is relevant, but around what a particular model can reliably use.

ETHICS STATEMENT

This work uses two publicly released benchmarks, BBEH and RealMemBench, under their respective licences, and introduces no new data collected from human subjects. The RealMemBench personas are synthetic conversational records distributed with the benchmark and contain no real user data, so no personally identifiable information is processed at any stage of memory construction, retrieval, or evaluation. All student, teacher, verifier, and judge roles are instantiated with a commercially available hosted language model, and we did not attempt to elicit, store, or redistribute training data from that model.

The method itself carries a dual-use consideration that we state explicitly. PUREMEM makes an external memory bank *model-specific* by probing a frozen model's empirical pass rate, which means the curation procedure encodes a measurement of that model's competence profile. Used as intended, this selects support material a model can exploit; the same measurement could in principle be used to characterise a deployed model's weaknesses. We report pass-rate statistics only in aggregate and release no per-item competence profile for any proprietary model. We declare no conflicts of interest and no sponsorship that influenced the design, analysis, or reporting of these experiments.

REPRODUCIBILITY STATEMENT

The complete configuration needed to reproduce our results is specified in the appendix rather than left to the released code. Section A.1 states the prompt-parity constraint $P_{\mathrm{mem}}(q, \varnothing) = P_0(q)$ that makes the no-memory and memory-augmented arms comparable; Sec. A.2 gives the model checkpoint used for all three roles, the split construction and random seed, and every decoding temperature and token budget; Sec. A.3 gives the ZPD probe count and band endpoints together with the resulting pass-rate distribution; Sec. A.4 gives the rejection-sampling budget, trace-acceptance criterion, embedding model, and the exact chunk counts of every memory bank we compare; Secs. A.5 and A.6 give all recall widths, matching bonuses, shortlist sizes, verifier sample counts, and thresholds for both tracks. Section A.7 defines every reported metric and states which baseline rows are reconstructed offline rather than executed as independent arms. Sections A.8 and A.9 specify the placebo intervention, the exclusion rule for truncated completions, and the exact McNemar, permutation, and bootstrap procedures, including the discordant counts needed to recompute the reported p-values. Source code, prompt templates, the constructed memory banks, and the per-query evaluation records are provided as anonymised supplementary material.

AI USE STATEMENT

Generative AI is a *research artefact* in this work: the frozen student model that is probed and evaluated, the teacher model that generates solution traces, the Stage-III applicability verifier, and the RealMemBench answer generator and judge are all instantiated with a hosted large language model, as described in Sec. A.2. All such uses are part of the method and the experimental protocol and are reported in full in the main text and appendix.

In addition, we used generative AI tools for writing assistance and for coding assistance. Writing assistance was limited to language editing of author-written drafts—grammar, concision, and consistency of terminology—and no section of this paper was drafted by a model from a prompt. Coding assistance was limited to routine implementation and analysis scripting; all experimental code, statistical procedures, and reported numbers were reviewed and re-run by the authors. We did not use generative AI for research ideation, for the design of the ZPD curation criterion or the three-stage retrieval architecture, for literature search or the generation of citations, or for producing any experimental result, figure, or table. We have reviewed all AI-assisted work and take full responsibility for the final content of this paper, including its text, claims, and released artefacts.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Haonan Bian, Zhiyuan Yao, Sen Hu, Zishan Xu, Shaolei Zhang, Yifu Guo, Ziliang Yang, Xueran Han, Huacan Wang, and Ronghao Chen. Realmem: Benchmarking llms in real-world memory-driven interaction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14349–14365, 2026.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Peng Cui and Mrinmaya Sachan. Investigating the zone of proximal development of language models for in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6485–6498, 2025.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman & Hall/CRC, 1994.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Yuanzhu Peter Chen, et al. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26473–26501, 2025.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International conference on learning representations*, 2020.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114, 2022.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5011–5034, 2023.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11048–11064, 2022.

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

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2655–2671, 2022.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International conference on machine learning*, pp. 31210–31227. PMLR, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Pragya Srivastava, Satvik Golechha, Amit Deshpande, and Amit Sharma. Nice: To optimize in-context examples or not? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5494–5510, 2024.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 14918–14937, 2023.

Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in neural information processing systems*, 36:74530–74543, 2023b.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33, 2020.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International conference on learning representations*, 2023c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *arXiv preprint arXiv:2203.08913*, 2022.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in neural information processing systems*, volume 35, 2022.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. Pmlr, 2021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. *Advances in neural information processing systems*, 36, 2023.

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