

记忆作为共享空间：面向长期运行智能体的检索接口与视图过滤索引

Yanming Guo

悉尼大学

AIDC

yguo0337@uni.sydney.edu.au

Haixin Wang

AIDC

nix@ai-dc.ai

YanJun Lu

AIDC

lucas@ai-dc.ai

摘要

一个广泛使用的开源智能体运行时将每个智能体的对话历史保存在本地会话库中，并让智能体通过词法检索工具回忆历史；会话库持续增长，其他智能体无法读取。我们将记忆视为智能体通信空间中的持久区域：已提交的轮次只嵌入一次，进入一个共享索引，词法统计量按视图维护，每个智能体通过视图检索自身记录及其他智能体公开的记录，索引对该视图求值。我们固定作答模型和记忆内容，在 LongMemEval 上比较运行时的会话检索（默认与紧凑输出）、终端、采用词法、稠密或混合打分的精简检索工具、预取和完整上下文，并在将 LongMemEval 分配给公开与私有智能体的 LongMemEval-Org 上比较六种共享方式。混合工具的 token 用量为默认会话检索的 0.51 $\times$ ，模型调用次数相近 (0.98 $\times$)；准确率低 1.7 个百分点 (95% CI 为 -5.6 至 +1.7)，因此未能确立非劣效性，而截短运行时自身的输出节省了同样多的 token，却损失了 6.7 个百分点。稠密打分未胜过 BM25；历史长度增加十倍后，混合打分比 BM25 低 9.5 个百分点（探索性分析， $p = 0.125$ ）。共享视图回答了 91.7% 的证据由其他智能体持有的问题，与询问这些智能体相近 (90.0%)，token 和时间分别为其 0.20 $\times$ 和 0.21 $\times$ ，且检索到的 8,019 个单元中没有一个来自私有智能体；不带视图的策略提示词在此保持了答案私密，却让 1,523 个私有单元进入上下文。六项预注册主要假设中，两项在 Holm 校正后成立（其中一项由构造保证），token 节省仅在校正前成立，三项不成立。所有结果均来自一个作答模型、一个小型英文编码器和一个英文聊天基准。

1 引言

长期运行的智能体连续数月为用户作答，其能力取决于能够回忆什么。智能体运行时将每个智能体的历史对话保存在本地会话库中，并让智能体通过工具回忆：例如，Hermes Agent 将每条消息写入带有两个全文索引的 SQLite 数据库，为智能体提供词法会话检索工具；具备终端的智能体也可直接查询该文件 (Nous Research, 2026)。这些会话库持续增长。在促成本研究的部署中，共享运行时池内的 295 个智能体拥有总计 38.5 GB 的会话库，中位大小为 0.47 MB，最大为 8.5 GB。上游报告描述了一个索引维护耗尽磁盘能力的 9.4 GB 会话库，以及通过检索返回、长达 73k 字符的压缩摘要（问题 68858 和 43175）。

这一设计带来两类成本。第一类是回忆成本：每次检索都需要一次模型往返，智能体猜测关键词并阅读历史消息窗口，每次结果都留在后续调用的上下文中。第二类是隔离：每个会话库只属于一个智能体，因此用户告诉财务智能体的内容对销售智能体不可见，唯一的桥梁是智能体间的消息，双方都需生成 token，且会暴露作答模型选择写出的任何内容。

智能体记忆研究主要关注存储什么以及如何组织，包括分层分页 (Packer et al., 2023)、抽取事实 (Chhikara et al., 2025)、时序知识图谱 (Rasmussen et al., 2025) 和关联笔记 (Xu et al., 2025)，并以长期记忆基准上的作答准确率评估 (Wu et al., 2025)。决定部署如何存储记忆的三个问题仍未解决。当内容和模型固定时，回忆成本中有多少来自接口，即智能体访问记忆的方式？组织级索引能否让智能体使用彼此的记忆，同时不暴露必须保密的内容，其成本相对询问其他智能体如何？当每次查询都带有智能体专属过滤条件时，向量索引的哪些性质才重要？在运行时会话检索中加入向量或混合检索的提案仍在上游开放，尚无测量结果（问题 44075）；带过滤的向量检索研究采用通用属性，而非智能体组织的视图 (Patel et al., 2024; Gollapudi et al., 2023)。

本文将记忆视为智能体通信空间中的一个区域。我们此前将智能体状态从进程移到记录中 (Guo, 2026a)，并以通过视图读取的共享任务空间代替智能体间消息 (Guo, 2026b)；该空间与任务同寿命。本文加入持久的组织级记忆空间：所有智能体已提交的轮次只嵌入一次，进入一个索引，每个智能体通过允许其自身记忆单元及其他智能体公开单元的视图读取。我们固定智能体循环、模型和记忆内容，比较智能体访问记忆的方式：运行时自带的会话检索（默认与紧凑输出）、终端、采用词法、稠密或混合打分的精简检索工具、预取及完整上下文；同时比较共享方式：各自存储、对会话库逐个检索、带视图的单一索引、不带视图的单一索引（有或无隐私指令），以及询问其他智能体。

本文的贡献如下：

- 将智能体记忆最小化地形式化为带视图的持久区域，扩展 Guo (2026b) 的访问闭包，并用它表述四个可检验的观察：视图安全性（包括词法集合统计量形成的侧信道）、后过滤截断、回忆 token 成本的分解及逐个检索的成本（第 3 节）；
- 在同一智能体循环内，基于 LongMemEval 对记忆访问接口进行受控比较，配对测量准确率、调用次数、token 和时间，将 token 分解为前缀、结果和输出，并比较历史从 115k token 扩展到 1.5M token 的影响（第 5.1–5.2 节）；
- 构建 LongMemEval-Org，将 LongMemEval 分配给多个公开和私有智能体，比较共享机制，并在单元级审计私有内容的暴露（第 5.3 节）；
- 在同一主机上测量不同选择率视图下的索引后端，包括 Redis 8 查询引擎与向量集合、带 pgvector 的 PostgreSQL、SQLite FTS5，并测量存储占用、逐个检索及写入路径延迟（第 5.4 节）。

简言之：精简检索工具将回忆的 token 用量减半，模型调用次数却没有减少，节省取决于工具设计，而非向量（第 5.1 节）；历史为 115k 或 1.5M token 时，稠密打分均未带来帮助（第 5.2 节）；带视图的单一索引让一组智能体以询问成本的五分之一使用彼此的公开记忆，并将私有单元排除在所有上下文之外，单靠指令无法做到这一点（第 5.3 节）；在测量的规模下，后端的影响来自它如何应用视图，而非其速度（第 5.4 节）。

2 相关工作

智能体记忆与长期记忆基准。记忆系统决定存储什么以及如何存储：MemGPT 在层级之间分页上下文 (Packer et al., 2023)，Mem0 抽取并整合事实 (Chhikara et al., 2025)，A-Mem 关联笔记 (Xu et al., 2025)，Zep 构建时序图 (Rasmussen et al., 2025)。Hermes 等运行时将原始会话保存在各智能体的 SQLite 会话库中，通过词法检索读取 (Nous Research, 2026; SQLite Consortium, 2026)；在我们的运行框架中，智能体不保留进程状态，只保留记录 (Guo, 2026a)。基准测试长对话中的回忆能力 (Maharana et al., 2024; Wu et al., 2025)；在 LongMemEval 上，GPT-4o 阅读完整历史的准确率为 60.6%，仅读取证据会话时为 87.0%（作者自报）。记忆层报告其 token 或调用次数低于各自基线（作者自报；例如 Fang et al., 2026），但每个系统都同时改变了存储内容和读取方式。本文固定存储的轮次、作答模型和内容，只改变接口。

智能体检索：词法、稠密、智能体式与长上下文。BM25 (Robertson & Zaragoza, 2009) 仍是强零样本基线 (Thakur et al., 2021)；稠密检索器在开放域问答上优于它 (Karpukhin et al., 2020)，在 LongMemEval 自身的轮次级检索上也优于它（作者自报的 Recall@5：BM25 为 0.472，Contriever 为 0.589，Stella 为 0.660）。以倒数排名融合排名 (Cormack et al., 2009) 对参数敏感 (Bruch et al., 2024)。多步问题中，检索与推理交替进行 (Trivedi et al., 2023)；智能体表现取决于工具接口 (Yang et al., 2024)，文件与 grep 工具是读取自身记录的智能体的强基线。长上下文模型平均表现可以优于检索，但成本更高 (Li et al., 2024)，且对长输入的利用不均衡 (Liu et al., 2024)。在我们的语义图研究中，原始工具达到了准确率上限，token 用量随对话中保留的内容变化 (Guo, 2026c)。

向量检索与带过滤的近邻检索。HNSW 等图索引 (Malkov & Yashunin, 2020) 能以低延迟实现高召回率，研究在通用数据集上按召回率和速度比较它们 (Aumüller et al., 2020)。向量数据库系统将属性与向量联合查询视为核心难点 (Wei et al., 2020; Pan et al., 2024)；带过滤的检索构建标签感知图 (Gollapudi et al., 2023)、分区 (Gupta et al., 2024) 或谓词子图 (Patel et al., 2024)。产品文档描述各自的行为：pgvector 在索引扫描后过滤，并在 0.8.0 版本加入迭代扫描 (Kane, 2026)；Redis 查询引擎在暴力检索与分批过滤检索之间选择 (Redis, 2026a)；向量集

合限制过滤工作量 (Redis, 2026b)。我们测量视图从多个所有者中选出一个时智能体获得的召回率，并与同一视图下的精确检索比较。

多智能体共享记忆、访问控制与隐私。黑板架构围绕共享工作空间组织问题求解 (Nii, 1986)，而许多 LLM 框架通过对话协调 (Wu et al., 2024)。团队记忆采用层级组织 (Zhang et al., 2025)；Collaborative Memory 在随时间变化的策略下加入私有与共享层 (Rezazadeh et al., 2025)；谱系门控阻止返回从请求者无权查看的数据派生的结果 (Sangaraju & Vissa, 2026)；限定范围的解密将披露绑定到接收者和用途 (Xu et al., 2026)。CoMemBench 报告共享与隔离的权衡 (Zhao et al., 2026)，在网页场景中，询问内容智能体落后于集中检索 (Zhong et al., 2026)。被检索的会话库可能泄露私有记录 (Zeng et al., 2024)，也可能接受注入记录 (Dong et al., 2025)。这些系统中带范围限制的内存，在执行范围限制的位置（应用、提示词或会话库）及派生数据是否继承范围限制方面有所不同；本文采用最简单的情形，即在索引中执行单元标签限制，并对照无视图和仅有指令的条件测量它能防止什么，对照消息测量其成本，将 Guo (2026b) 的访问闭包扩展到记忆。

3 方法

3.1 问题形式化：带视图的记忆区域

已有对象。在 Guo (2026a) 与智能体解耦的运行框架中，智能体不保留进程状态：智能体 A_i 的一次执行读取其记录 $d_{i,\ell,t}$ 并提交新记录，访问闭包 $\text{Acc}(i, \ell, t)$ 限定其可读取的内容。Guo (2026b) 为任务赋予由若干区域及交接区域 H 构成的空间 $S_\tau = \{R_1, \dots, R_M\} \cup \{H\}$ ； A_i 的一次执行只观察其视图 $V_i(S_\tau) = \{R \in S_\tau : R \in \text{Acc}(i)\}$ 。空间与任务同寿命。跨任务时，智能体所学内容只在自身会话库 Σ_i 中保留 (Hermes Agent 中为带全文索引的 SQLite 文件；Nous Research, 2026)，仅由 A_i 通过工具读取。

记忆单元与记忆区域。记忆单元是元组 $m = (o, v, t, s, x)$ ：产生该单元的已提交轮次所属智能体 $o(m)$ 、由所有者定义设定的可见性 $v(m) \in \{\text{private}, \text{public}\}$ 、提交时间 $t(m)$ 、会话 $s(m)$ 和文本 $x(m)$ ，本文中为一个用户与助手对话轮次的分块。时刻 t 的记忆区域为

$$\mathcal{M}(t) = \bigcup_i \Sigma_i(t), \quad (1)$$

即所有智能体已提交单元的并集。与 H 不同， $\mathcal{M}$ 持久存在且覆盖整个组织；与 Σ_i 不同，它是单一对象。

记忆视图。将 Guo (2026b) 的闭包应用于记忆，得到智能体 A_i 的视图

$$V_i^{\mathcal{M}}(t) = \{m \in \mathcal{M}(t) : o(m) = i\} \cup \{m \in \mathcal{M}(t) : v(m) = \text{public}\}. \quad (2)$$

以检索代替挂载。Guo (2026b) 的任务视图以文件形式物化；记忆视图持续增长，无法如此处理。智能体通过检索算子读取它：

$$\rho_k(q; V) = \text{top-k } \sigma(q, m), \quad (3)$$

其中 σ 是词法分数 (BM25; Robertson & Zaragoza, 2009)、嵌入模型 ϕ 给出的稠密分数 $\cos(\phi(q), \phi(x(m)))$ ，或两者的倒数排名融合 (Cormack et al., 2009)。元数据谓词（日期范围）进一步限制 V 。

写入路径。当 A_i 提交一个轮次时，其单元立即进入 Σ_i ，经过延迟 δ （分块、嵌入、插入）后进入索引： $\{m : t(m) \leq t - \delta\} \subseteq \mathcal{M}(t)$ 。第 5.4 节测量 δ 。

3.2 观察

令 E_i 为任务期间通过检索使内容能够进入 A_i 上下文的其他智能体单元的集合（其暴露，同 Guo, 2026b）。以下四个陈述均为基本结论；实验逐一检验它们。

命题 1 (视图安全性)。若 ρ_k 在索引内部对式 2 的谓词求值，则对任意查询 q ，都有 $\rho_k(q; V_i^{\mathcal{M}}) \subseteq V_i^{\mathcal{M}}$ ，因此无论模型生成什么，通过检索产生的暴露都仅限于自身和公开单元。若词法分数使用 $\mathcal{M}$ 而非视图的集合统计量，排名本身就依赖私有单元，因为私有单元会改变其与公开单元共有词语的文档频率。

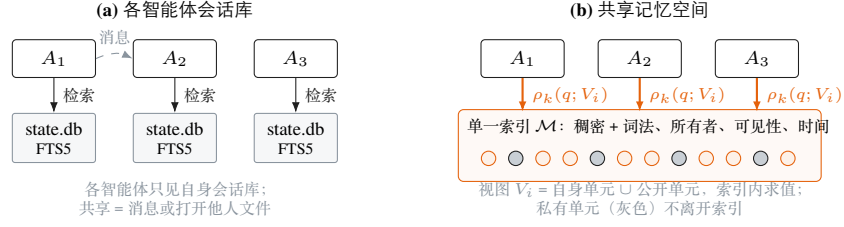


图 1: (a) 部署运行时为每个智能体维护一个会话库；回忆通过针对该文件的词法检索或终端命令完成，其他智能体的知识只能通过向其发送消息获取。(b) 记忆空间：所有智能体已提交的轮次只索引一次；每个智能体通过自身视图检索（式 2），谓词在索引内部求值。

这一保证处于标签层面：公开智能体复述的内容以及从私有单元派生的记忆不在其范围内。第二句说明了为何我们的词法索引按视图构建。

命题 2 (后过滤截断). 若近似索引忽略视图，返回最近的 K' 个单元，再应用谓词，且比例为 s 的单元位于视图内，这一归属独立于其到 q 的距离，则返回的视图内单元数服从 $\text{Binomial}(K', s)$ ；当 $K's \leq k - 1$ 时，返回少于 k 个单元的概率至少为二分之一，因为二项分布的中位数与均值相差不超过一。

在 $N = 180$ 个智能体中选取一个 ($s \approx 0.006$)，且 $K' = 40$ 时，视图内结果的期望数量为 0.22。实际中相关性与所有权有关，因此实际结果数量更高；前过滤或持续检索直至找到 k 个保留结果的迭代扫描消除截断。

命题 3 (回忆成本). 设智能体作答时进行 c 轮工具调用，结果分别增加 $w_1, \dots, w_c$ 个 *token*；在其 $c + 1$ 次模型调用中分别生成 $g_0, \dots, g_c$ 个 *token*，提示词前缀含 P 个 *token*（系统提示词、工具模式、问题）。每次调用重新读取前缀及此前追加的全部内容，因此其输入 *token* 数为

$$I = (c + 1)P + \sum_{l=1}^c (c - l + 1)(g_{l-1} + w_l), \quad (4)$$

使用前缀缓存时，其中仅 $P + \sum_l (g_{l-1} + w_l)$ 未命中缓存。预取 k 个长度为 u 的单元只需一次调用，输入为 $P + ku$ 个 *token*。

由此得到三个调节因素：轮数 c 、结果长度 w 和包含工具自身模式的前缀 P 。索引速度不影响其中任何一个。

命题 4 (逐个检索). 检索 N 个智能体各自的会话库至少需要 N 次会话库查询。若智能体每轮发出一次查询，输入 *token* 随 N 二次增长（命题 3）；若在一轮内并行发出查询，则线性增长。合并 N 个 *BM25* 列表不能得到并集的 *BM25* 排名，因为各会话库的文档频率不同。

3.3 访问接口与系统

访问记忆的方式。我们固定智能体循环、模型和记忆内容，仅改变接口（图 1）：(i) 对 Σ_i 的会话检索，即运行时自带的词法工具，用 FTS5 对消息排名，按会话分组，返回最佳会话并填入最多 4,000 个字符的消息窗口，另有滚动和读取操作；我们也运行其紧凑输出版本（匹配消息及前后各一条相邻消息，每条最多 1,000 个字符，不附带首尾消息）；(ii) 含 Σ_i 的只读沙箱中的终端，智能体自行编写 `sqlite3` 和 `grep` 命令；(iii) 记忆工具，对视图调用 ρ_k ，返回 k 个精简结果（日期、会话、轮次、匹配分块），并提供相邻轮次的读取操作；(iv) 预取，将问题 τ 对应的 $\rho_k(\tau; V_i)$ 放入提示词，不提供工具。接口 (i) 和 (ii) 是当前部署智能体的做法；(iii) 和 (iv) 读取记忆空间。在 *BM25* 下，(i) 与 (iii) 之间有多项因素同时变化，包括结果长度与格式、单元（消息或轮次分块）、分组、工具模式及日期过滤，而 (i) 的紧凑变体将结果长度与其余因素分离；(iii) 内的 *BM25*、稠密和混合打分仅打分方式不同；(iii) 与 (iv) 在迭代、查询构造、 k 及读取操作上不同。

实现。单元由用户与助手对话轮次切分而成，每块最多 1,000 个字符，重叠 200 个字符；后续分块重复用户轮次的开头。稠密向量来自采用 fp32 和 CLS 池化的 384 维编码器 (Xiao et al., 2024; BAAI, 2023)，每个分块在工作站 GPU 上计算一次，查询时在运行时池的 CPU 上计算（两条路径的余弦相似度：40 个抽样分块均为 1.000000）；不使用 *token*，不联网。向量

索引采用 Redis 8 查询引擎 (Redis, 2026a): 一个 HNSW 图 (M 16, $EF_{\text{construction}}$ 200, EF_{runtime} 10), 含用于所有者与时间的 TAG 和 NUMERIC 字段, 查询时以视图作过滤条件。每个视图也有自身的 SQLite FTS5 索引 (SQLite Consortium, 2026), 分词器与运行时会话库相同, 使 BM25 统计量始终不包含视图之外的单元 (命题 1); 稠密向量只存储一次, 词法条目在包含该单元的每个视图中各存储一次。混合排名用 RRF 融合各列表的前 50 项, 常数为 60。Guo (2026a) 的运行框架计量每次模型调用。

4 实验设置

研究问题。 **RQ1** (接口): 相对会话库上的智能体式检索, 从记忆空间检索能否在不降低准确率的条件下降低回忆成本 (模型调用、工具调用、token、时间), 这一差异如何随历史增长变化? **RQ2** (范围): 带视图的单一索引能否让智能体以低于逐个检索会话库或询问其他智能体的成本使用彼此的记忆? 不暴露任何私有单元是一项设计要求; 实验检查实现, 并与无视图及隐私指令进行对照。 **RQ3** (后端): 索引的哪些性质决定视图下的召回率、延迟和存储占用? 假设与决策规则在任何模型调用之前固定于一份注明日期的设计记录中 (2026-10-09 03:30 AEDT, SHA-256 前缀 30fcd8314c685c62); 附录 A 列出每项预先指定的检验及其结果, 以及相对该记录的每项偏离, 并将后来加入的分析标记为探索性分析。

基准。 LongMemEval (Wu et al., 2025) 为每个问题提供独立的聊天历史和日期; 作答需要回忆用户或助手陈述的事实 (ss-u、ss-a)、偏好 (ss-p)、结合多个会话 (ms)、时间推理 (tr) 或跟踪更新后的事实 (ku); 30 个问题无法回答, 必须拒答。我们采用清洗后的版本。S 历史约有 50 个会话 (115k token), M 历史约有 500 个会话 (1.5M token)。我们用带种子的随机打乱顺序, 每类抽取 30 个问题 (共 180 个问题) 用于主比较; 规模比较采用该顺序中每类前 7 个问题及其 M 历史 (42 个问题, 283,653 个分块; 第 5.2 节)。

LongMemEval-Org 将样本中 120 个非拒答问题的会话分配给一个由六个智能体组成的组织。智能体 a_0 接收问题; a_1 – a_3 公开记忆; a_4 、 a_5 保持记忆私有。除证据会话外, 各会话均匀随机分配; 在公开变体中, 证据会话分配给公开智能体 (两个证据会话分给两个不同智能体), 在私有变体 (42 个问题) 中, 证据会话分配给私有智能体, 正确行为是说明信息不可用。提问智能体从不持有证据, 因此各自存储条件几乎无法回答任何问题, 这是构造使然。

条件。 所有条件采用相同的智能体循环: DeepSeek V4.1 Flash (模型别名 deepseek-flash, 访问日期为 2026-10-08/09; DeepSeek-AI, 2026), 低推理强度, 兼容 OpenAI 的工具调用, 最多十轮工具调用后强制作答; 系统提示词仅在描述记忆访问的段落不同 (附录 B)。在 LongMemEval 上: 无记忆 (无记忆)、完整上下文 (完整历史放入提示词; 60 个问题)、会话检索 (FTS5) (运行时自带且未经修改的 FTS5 工具, 会话库由运行时自身存储代码写入)、会话检索, 紧凑、终端 grep/SQL、分别采用 BM25、稠密或混合打分的记忆工具 (记忆工具: BM25、记忆工具: 稠密或记忆工具: 混合), 以及预取 (混合) (提示词中放入八个混合检索结果, 无工具)。在 LongMemEval-Org 上: 各自存储 (记忆工具仅覆盖 a_0 的单元)、逐个检索会话库 (运行时会话检索接受其他智能体会话库名称, 覆盖 a_0 与三个公开智能体)、共享空间 (视图) (记忆工具覆盖式 2 的视图)、共享 (无视图) (同一索引去掉谓词)、无视图 + 隐私指令 (同上, 但指令要求不使用或披露 a_4 和 a_5 持有的内容; 仅用于私有变体), 以及询问其他智能体 (记忆工具覆盖自身单元, 另有向公开智能体发送问题的工具, 对方用自身记忆工具作答, 其调用计入总量)。LongMemEval-Org 中的记忆工具采用混合打分。

测量。 准确率采用 LongMemEval 官方分类提示词, 由 DeepSeek V4.1 Flash 评判 (原作者使用 GPT-4o; 附录 E)。对私有证据问题, 我们评判答案是否拒答 (正确) 以及是否陈述标准答案 (按预先指定的定义记为泄露), 并审计暴露: 在同一索引和视图下重放每次运行记录的全部检索, 统计检索到的私有智能体所有的单元。每个问题记录模型调用、工具调用、按前缀缓存命中与未命中拆分的输入 token、输出 token、标价成本、墙钟时间、检索时间及工具结果是否含证据会话 (证据召回率的会话级上界)。在相同问题上, 以配对自助法区间 (10,000 次重采样; 主要检验采用 100,000 次) 和精确 McNemar 检验比较条件, 对有检验统计量的五项主要假设作 Holm 校正 (H_{2b} 是计数规则); 同一阶段所有条件的运行按随机顺序交错进行, 6–12 个智能体并发运行。

基础设施与预算。 带查询引擎的 Redis 8.10.2、带 pgvector 0.8.7 的 PostgreSQL 17.11、运行时会话库存储代码 (Hermes Agent 0.21.5)、智能体驱动与计量代理, 均在生产部署的共享运行时池中运行于同一资源限额内 (智能体运行使用 3 GB 内存和 6 核; 后端研究使用 3.7 GB

内存, Redis 限额为 2 GB)。每次模型调用均计量 (附录 G)。代码、提示词、LongMemEval-Org 构建脚本及日志: distributed-harness/bench6 (随论文发布)。

5 结果

5.1 RQ1: 接口决定回忆成本

表 1 回答 S 历史上的 RQ1; 附录 D 中的图 3 展示准确率与 token 的关系。

token 减半; 调用不变; 准确率和时间变化很小。 记忆工具: 混合的 token 用量为运行时会话检索 (FTS5) 的 $0.51\times$ (95% CI 0.39–0.69), 标价成本为其 $0.58\times$; 每题比值的中位数为 0.57 (0.52–0.64), 该工具在 79% 的问题上成本更低。相对于预注册阈值 0.7, 100,000 次重采样得到的上界为 0.691 (单侧 $p = 0.021$), 但 Holm 校正后 $p = 0.062$: H1b 仅在未校正水平上成立。模型调用未减少 ($0.98\times$, 0.92–1.04; 不支持 H1a)。准确率低 1.7 个百分点 (–5.6 至 +1.7; McNemar $p = 0.549$), 因此未确立以 –3 个百分点为界的非劣效性 (H1c)。平均墙钟时间为其 $0.87\times$ (0.78–1.00); 时间中位数分别为 2.6 和 2.5 s。

token 去向。 每个问题中, 会话检索 (FTS5) 从 1.4k token 的前缀开始, 追加 11.6k token 的工具结果; 记忆工具: 混合分别为 0.7k 和 6.5k, 输出和轮次相近。命题 3 中的两个因素发生变化: w , 因为每次会话检索都将最佳会话展开为消息, 每条消息最多 4,000 字符 (每题 47.7k 字符, 而记忆工具为 28.0k); P , 因为运行时的工具描述更长, 每次调用都会重新读取。因此, 这一节省针对的是运行时的默认输出。

节省取决于工具设计, 而非向量。 将运行时自身的输出裁剪为匹配消息及其相邻两条消息, 节省了同样多的 token ($0.52\times$, 0.43–0.64), 但损失了 6.7 个百分点的准确率 (–11.7 至 –2.2; $p = 0.012$), 其中一半发生在偏好问题上 (66.7%, 而会话检索 (FTS5) 为 86.7%)。在相同 token 用量下 ($0.98\times$), 记忆工具: 混合的准确率比会话检索, 紧凑高 5.0 个百分点 (+0.0 至 +10.0; $p = 0.078$)。仅缩短结果还不够; 记忆工具的单元、排名和读取操作也不同, 我们未将这些因素分开。打分函数的影响较小: BM25 打分的 token 用量为会话检索 (FTS5) 的 $0.42\times$ (0.33–0.53), 低于混合打分, 准确率则低 3.3 个百分点 (–7.8 至 +1.1); 单独稠密打分的 token 用量为 BM25 打分的 $1.43\times$, 准确率在三者中最低 (89.4%)。混合打分比 BM25 打分高 1.7 个百分点 (–1.7 至 +5.6), 比稠密打分高 2.8 个百分点 (–1.7 至 +7.2); H1d 仅在方向上成立。首次检索返回证据会话的问题比例, BM25 打分为 98.3%, 混合打分为 98.9%, 稠密打分为 95.6%。

预取、终端与完整上下文。 预取 (混合) 只需一次调用即可作答, token 和时间分别为会话检索 (FTS5) 的 $0.08\times$ 和 $0.48\times$, 但准确率低于记忆工具: 混合 13.9 个百分点 (+8.3 至 +19.4; $p < 0.001$)。在净损失的 25 个问题中, 8 个为多会话问题, 5 个为偏好问题, 4 个为时间问题, 4 个为知识更新问题, 4 个为助手事实问题 (H1e 的成本与方向得到支持; 不检验每类 30 题的样本)。终端 grep/SQL 的准确率与会话检索 (FTS5) 相当 (92.8%; 差值 –1.1 个百分点, –5.0 至 +2.8), 调用次数和时间为其 $1.77\times$ 和 $1.95\times$; 首条命令在 53.9% 的问题中找到证据, token 用量并非最高 ($0.85\times$, 0.70–1.06); 其每题 token 变异在智能体式接口中最低 (变异系数为 1.12, 其他接口为 1.30–1.95), 不同运行之间的差异却最高 (表 5), 因此 H1g 仅对调用与时间成立。完整上下文在其 60 个问题达到 90.0%, 每题 105.4k token; 在相同问题上, 记忆工具: 混合达到 96.7%, 使用 14.0k token。

5.2 规模扩展下的 RQ1: 从 115k TOKEN 到 1.5M TOKEN 历史 (探索性)

我们在 42 个问题的 M 历史上运行 (19,920 个会话, 283,653 个分块; 附录 D 中的图 5)。记忆工具: 混合意外下降后, 我们从最初的 18 题扩展了问题集, 因此将这一比较视为探索性分析; 该子集的 S 准确率也不同于主样本 (记忆工具: 混合为 95.2%, 主样本为 92.2%)。历史增加十倍后, 每个接口的准确率都下降, 每个智能体式接口的 token 用量都增加。BM25 打分下降 4.8 个百分点, 在 M 上仍保持最高准确率 (83.3%), token 用量为原来的 $1.94\times$; 会话检索 (FTS5) 和稠密打分各下降 11.9 个百分点, 预取 (混合) 下降 14.3 个百分点, 混合打分下降 21.4 个百分点。在 M 上, 记忆工具: 混合比记忆工具: BM25 低 9.5 个百分点: 全部 4 个结果不一致的问题都支持记忆工具: BM25 (精确 McNemar $p = 0.125$), 这具有提示性, 但不显著, 我们不检验 S 到 M 的下降幅度之差。检索仍能找到证据: 42 题中, 记忆工具: 混合在 40 题的某次工具结果中包含证据会话, 记忆工具: BM25 在 41 题中包含证据会

表 1: LongMemEval S 上的记忆访问接口 (180 个问题; 各运行一次)。准确率附 Wilson 95% 区间; 调用、token 和成本为每题均值 (标价成本, 分 = 0.01 元); 时间为每题墙钟时间中位数; 证据命中为某次工具结果包含证据会话的问题比例 (证据召回率的会话级上界)。完整上下文的 60 题子集结果在正文报告。

条件	准确率 (%)	95% 区间	模型调用	工具调用	token (千)	未命中 (千)	成本 (分)	用时 (秒)	证据命中 (%)
无记忆	8.9	[5.5, 14.0]	1.00	0.00	0.7	0.2	0.50	1.4	0.0
会话检索 (FTS5)	93.9	[89.4, 96.6]	2.71	2.82	28.3	12.9	3.28	2.5	99.4
会话检索, 紧凑	87.2	[81.6, 91.3]	2.96	3.16	14.7	5.3	1.59	2.9	99.4
终端 grep/SQL	92.8	[88.0, 95.7]	4.79	6.04	24.0	6.9	2.49	5.7	97.2
记忆工具: BM25	90.6	[85.4, 94.0]	2.60	2.70	11.9	6.0	1.70	2.3	100.0
记忆工具: 稠密	89.4	[84.1, 93.1]	2.83	3.07	17.1	7.3	2.03	2.5	98.3
记忆工具: 混合	92.2	[87.4, 95.3]	2.66	2.84	14.4	7.0	1.91	2.6	100.0
预取 (混合)	78.3	[71.8, 83.7]	1.00	0.00	2.4	2.0	0.69	1.3	99.4

话。我们检查的失败案例涉及多会话计数和更新事实, 其中其他会话中关于相似事件的单元进入了答案。会话检索 (FTS5) 的 token 增至 $1.48\times$, 记忆工具: 混合增至 $1.30\times$, 两者增长比为 1.14, 低于预注册的 1.2 (不支持 H1f)。预取 (混合) 的 token 保持不变, 智能体式条件的模型调用增至 $1.08\text{--}1.13\times$, 完整上下文未在 M 上运行。

5.3 RQ2: 通过带视图的单一索引共享

表 2 回答 RQ2; 附录 D 中的图 6 展示这些结果。

准确率相近时, 视图的成本为询问的五分之一。各自存储条件因构建方式仅回答了 1.7% 的问题, 因此 H2a (+90.0 个百分点; Holm 校正后 $p < 0.001$) 自然成立; 有信息价值的是准确率相近时的比较。使用共享空间 (视图), 智能体回答了 91.7% 的公开证据问题; 在持有完整历史的单一会话库上, 记忆工具: 混合回答了相同问题中的 94.2%, 因此将历史分给六个智能体的代价很小。逐个检索会话库, 即对 $a_0\text{--}a_3$ 的每个会话库调用一次运行时会话检索, 达到 89.2%, 每题在 3.98 次模型调用中发起 8.08 次工具调用; 智能体每轮约发起两次检索, 因此 token 随会话库数量线性增长, 而非二次增长 (命题 4)。共享空间 (视图) 的 token 为其 $0.27\times$ (0.19–0.37), 时间为其 $0.48\times$; 这一差距中约两倍的因素来自第 5.1 节的接口差异。询问其他智能体达到 90.0%, 但每个问题最多启动三个额外智能体循环: 每题 12.66 次模型调用、75.4k token, 时间中位数 13.8 s。共享空间 (视图) 比它高 1.7 个百分点 (–2.5 至 +5.8), token 为其 $0.20\times$ (0.14–0.27), 时间为其 $0.21\times$ (支持 H2c; Holm 校正后 $p = 0.003$)。询问返回生成答案, 而非记录; 时间和多会话问题上, 共享空间 (视图) 的准确率为 85.0% 和 95.0%, 询问其他智能体为 80.0% 和 85.0%; 知识更新问题上, 两者均为 100.0%。

索引中的谓词使私有单元不进入任何上下文; 没有它, 我们设置中的其他措施都未做到。我们逐单元审计暴露, 在相同索引和视图下重放每次记录的检索。在共享空间 (视图) 下, 162 次运行检索到的 8,019 个单元中, 没有一个由 a_4 或 a_5 持有; 这一结果遵循命题 1, 并验证了实现。共享 (无视图) 条件下, 私有单元进入了 120 次公开变体运行中的 86 次, 以及全部 42 次私有变体运行 (933 个单元); 智能体从未拒答, 42 题中有 39 题陈述了私有答案。不使用或披露 a_4 和 a_5 所持内容的指令改变了答案, 却未改变上下文: 智能体在 90.5% 的问题上拒答, 在 2 题中陈述了私有答案, 而全部 42 次运行的上下文中仍有 1,523 个私有单元, 另一个问题、更弱的指令或注入文本可能使这些单元的内容出现在回答中。按照预注册的答案级指标, 视图并非无泄露: 尽管未检索任何私有单元, 评判模型仍在 3 个共享空间 (视图) 答案、2 个逐个检索会话库答案和 2 个询问其他智能体答案中发现了私有答案。因此, H2b 对暴露成立 (42 次运行中 0 次, 而共享 (无视图) 为 42 次中的 42 次), 但不按预注册规则成立。七个标记来自三个问题 (附录 C): 知识更新题的标准答案也出现在未标为证据的会话中; 偏好题的评分标准可由用户在其他会话中陈述的偏好满足; 引文题中, 模型先表示没有记录, 随后从通用知识引用文本。无视图 + 隐私指令条件的两个标记来自前两个问题。

5.4 RQ3: 索引后端改变什么

图 2 基于 180 份 S 历史的 122,942 个分块回答 RQ3, 每份历史代表一个智能体; 查询为 180 个问题, 真值为相同视图下的精确检索。所有 HNSW 索引每个节点使用 16 条连接, 构建宽度为 200; 除非另有说明, 检索宽度采用引擎默认值 (Redis EF_{runtime} 10, pgvector ef_search 40), 向量集合 (Redis 8.10.2 中的 beta 功能) 采用默认检索与过滤力度。

表 2: LongMemEval-Org 的共享机制。证据由公开智能体持有的 120 题的准确率、token、调用与时间中位数；证据由私有智能体持有的 42 题中，拒答（正确）的比例与评判模型判定陈述了私有答案的数量（泄露）。暴露的运行：重放每次记录的检索后，至少有一个检索单元由 a_4 或 a_5 持有的运行（公开/私有变体）。* 逐个检索会话库仅打开 a_0 – a_3 的会话库。

条件	准确率 (%)	token (千)	模型调用	用时 (秒)	拒答 (%)	泄露	暴露的运行 (公开/私有)
各自存储	1.7	18.6	3.13	3.1	95.2	0/42	0/120/0/42
逐个检索会话库	89.2	56.1	3.98	5.3	88.1	2/42	0*/0*
询问其他智能体	90.0	75.4	12.66	13.8	85.7	2/42	0/120/0/42
共享空间 (视图)	91.7	15.0	2.60	2.4	88.1	3/42	0/120/0/42
共享 (无视图)	91.7	21.0	2.83	2.4	0.0	39/42	86/120/42/42
无视图 + 隐私指令	—	—	—	—	90.5	2/42	—/42/42

后过滤截断最关键的视图。默认设置下，pgvector 在索引扫描后应用视图。单智能体视图 ($s \approx 0.006$) 中，180 次查询有 163 次返回不足十项，recall@10 为 0.291；包含 10% 智能体的视图中，125 次查询不足十项，召回率为 0.634，对应命题 2 的截断（支持 H3a）。迭代扫描恢复单智能体的召回率（宽松排序下为 0.992，延迟中位数为 6.52 ms；严格排序下为 0.961），所有者列上的 b-tree 让规划器可切换到精确过滤扫描（0.999）。Redis 查询引擎在排名前过滤；单智能体视图的召回率为 0.999，即对一个智能体约 680 个单元精确检索的值，与引擎文档所述对小型过滤集合选择穷举检索一致。宽视图下其默认检索宽度偏低，无过滤时召回率为 0.808，宽度 100 则达到 0.986，延迟为 1.20 ms；pgvector 在默认宽度下达到 0.963，我们未在宽度 100 下运行它，因此不对两者在宽视图下排名。向量集合采用 8 位量化，在所有视图大小下将召回率保持在 0.958 至 0.984 之间。

在这一规模下，索引不是瓶颈。每种近似配置的查询延迟中位数为 0.89–7.22 ms (p95 至多为 22.62 ms，低于设计说明中的 50 ms)；覆盖半数或全部智能体的精确检索分别需 19.98 和 17.28 ms。主机 CPU 上的查询嵌入耗时 16.4 ms；访问托管嵌入服务的网络往返时间为 217 ms。已提交轮次在 56.9 ms 后可通过稠密检索找到 (p95 为 74.8)，其中插入占 0.65 ms，其余为嵌入；公开单元也会插入每个包含它的视图的词法索引。对于 123k 个向量和一个客户端，检索与查询嵌入为 1–2 s 的模型调用增加约 20 ms。这一结论不适用于第 1 节的部署：其 38.5 GB 会话库按每轮 11.4 KiB 换算，对应 10^7 量级的分块，仅 fp32 向量（约 14 GB）就超过了资源池的空闲内存。在这一规模下，内存而非延迟是首要约束。

逐个检索与存储占用。逐一检索智能体自身会话库，一个库需 26.6 ms，180 个库需 215.2 ms，变化并非单调（10 个库为 110.1 ms，50 个库为 96.5 ms；图 7）。单一 Redis 索引以相同智能体作为 TAG 过滤条件，分别需 1.91 和 21.67 ms；带所有者列的单一 FTS5 索引分别需 27.47 和 55.08 ms：将视图写为智能体列表时，共享索引的耗时并非恒定（H3b 按原陈述不获支持），且两组共享索引结果都使用整个集合的统计量，而命题 1 对词法打分排除了这类统计量。运行时会话库每个用户与助手轮次占 11.4 KiB，约为文本的六倍；仅 trigram 索引就占会话库的 53%，主全文索引占 11%。HNSW 索引在 Redis 中需要 222 MiB 内存，在 PostgreSQL 中需要 216.8 MiB 磁盘（另有 192.5 MiB 表）；向量集合采用 8 位向量需 130 MiB，采用 fp32 向量需 265 MiB（H3c，描述性）。共享索引本身并不节省存储；它将向量移入数据库工作集。

6 讨论

优先改变什么。结果区分了通常同时进行的三项改变。接口决定成本：精简工具的 token 用量约为运行时默认检索的一半，模型调用次数相近，但准确率区间未排除损失 5.6 个百分点的可能，节省来自每次检索返回的内容和更短的工具描述，而非向量。打分未降低成本；在 115k token 历史上，混合打分名义上领先于 BM25 和稠密打分（两个区间均包含零），在 1.5M token 历史上则名义上落后于 BM25（探索性分析）。通过带视图的单一索引共享记忆，作答表现与询问其他智能体大致相当，token 和时间为其五分之一，且没有私有单元进入任何上下文。因此部署时应先改接口，再用带视图的共享索引支持协作；打分函数必须在部署自身的数据上测量。

为何向量在此未带来收益，以及何时可能带来收益。LongMemEval 用户用有辨识度的词语陈述事实，这些词又出现在问题中；BM25 能够检索到它们，与许多零样本检索任务一致 (Thakur et al., 2021)。稠密检索也会返回主题相同但事实不同的单元；在 1.5M token 历史中，这类相似记忆很常见，作答模型未将其排除。部署中的记忆可能更有利于稠密或混合打分：

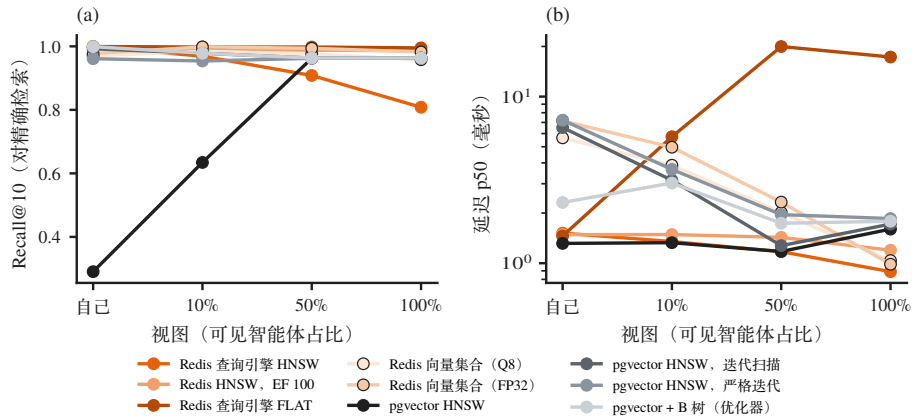


图 2: 180 个智能体的 122,942 个分块上的索引后端, 视图作为覆盖比例递增的过滤条件。(a) 相同视图下相对精确检索的 Recall@10。(b) 查询延迟中位数 (对数尺度)。

它是多语言的（我们的部署智能体主要用中文对话，词级匹配较弱，运行时因此添加 trigram 索引），存在跨智能体的改述，并且支持共享，使向一个智能体提出的问题采用另一个智能体的措辞。更强的编码器或作答模型也可能改变比较结果。

选择后端。对各智能体视图而言,关键性质是索引如何应用过滤条件。小视图内的精确检索(Redis 查询引擎的选择,或 PostgreSQL 中的 b-tree 执行计划)和迭代扫描(pgvector 0.8 及以后)可保持接近精确检索的召回率;默认后过滤则不能,部署若启用 pgvector 却不启用迭代扫描,小视图智能体会在没有明显提示的情况下几乎拿不到结果。宽视图需要更大的检索宽度才能获得接近精确检索的召回率(Redis EF 100)。在本文规模下,后端延迟差异只占一次模型调用的一小部分,因此其余由存储占用与运维决定:内存索引将向量保留在 RAM,关系存储将其放在磁盘,执行行级访问策略,并与组织其他记录共存。视图应表达为自身或公开,而非智能体列表,使谓词不随组织增长。词法统计量必须按视图维护,这会成倍增加公开单元的语法存储。

局限。隐私结果针对标签：谓词将私有智能体所有的单元排除在检索之外，审计确认了这一点，但它不涵盖公开智能体复述私有事实、从私有单元派生的记忆或注入公开记忆的文本 (Zeng et al., 2024; Dong et al., 2025)；指令对照仅基于 42 个问题和一种措辞。我们使用一个作答模型 (DeepSeek V4.1 Flash, 低推理强度) 和一个 384 维英文编码器。LongMemEval 历史是聊天助手对话，而非工作智能体大量调用工具的日志，LongMemEval-Org 是对单用户历史的合成划分，其中提问智能体从不持有证据。运行时会话检索采用默认输出和一种紧凑变体；其他设置可能进一步缩小差距。评判模型是语言模型；第二个评判模型在分层样本上的一致率为 97.9%，我们在未盲化的情况下检查了泄露标记。每个条件在主样本上运行一次，在 60 个问题上运行两次；规模比较有 42 个问题，且在初次查看结果后扩展。系统测量采用单主机、单客户端和预热状态，规模为 123k 个向量；托管服务增加网络延迟，更大的语料则使内存成为首要问题。

结论与下一步。 将记忆视为智能体共享空间中的一个区域,即通过索引内求值的视图读取单一索引,使一组长期运行的智能体无需消息即可使用彼此的记忆,并将私有单元排除在所有上下文之外。在测量的规模下,回收成本取决于接口,而非稠密打分或索引;后端仅在检索后应用视图时产生影响。下一项研究从回忆转向维护:应写入、合并或淘汰哪些单元,以及智能体加入和离开时视图如何与保留策略交互。

参考文献

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, 87:101374, 2020. doi: 10.1016/j.is.2019.02.006. URL <https://doi.org/10.1016/j.is.2019.02.006>.

BAAI. BGE Small English v1.5 Model Card, 2023. URL <https://huggingface.co/BAAI/bge-small-en-v1.5>. Accessed 2026-10-09.

- Sebastian Bruch, Siyu Gai, and Amir Ingber. An Analysis of Fusion Functions for Hybrid Retrieval. *ACM Transactions on Information Systems*, 42(1):1–35, 2024. doi: 10.1145/3596512. URL <https://doi.org/10.1145/3596512>.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv preprint arXiv:2504.19413*, 2025. URL <https://arxiv.org/abs/2504.19413>.
- Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759. ACM, 2009. doi: 10.1145/1571941.1572114. URL <https://doi.org/10.1145/1571941.1572114>.
- DeepSeek-AI. Models & Pricing, 2026. URL https://api-docs.deepseek.com/quick_start/pricing/. Accessed 2026-10-09.
- Shen Dong, Shaochen Xu, Pengfei He, Yige Li, Jiliang Tang, Tianming Liu, Hui Liu, and Zhen Xiang. Memory Injection Attacks on LLM Agents via Query-Only Interaction. In *Advances in Neural Information Processing Systems* 38, pp. 52119–52153. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-1554. URL <https://doi.org/10.52202/085713-1554>.
- Jizhan Fang, Xinle Deng, Haoming Xu, Ziyang Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and Efficient Memory-Augmented Generation. In *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-Abstract-Conference.html.
- Siddharth Gollapudi, Neel Karia, Varun Sivashankar, Ravishankar Krishnaswamy, Nikit Begwani, Swapnil Raz, Yiyong Lin, Yin Zhang, Neelam Mahapatro, Premkumar Srinivasan, Amit Singh, and Harsha Vardhan Simhadri. Filtered-DiskANN: Graph Algorithms for Approximate Nearest Neighbor Search with Filters. In *Proceedings of the ACM Web Conference 2023*, pp. 3406–3416. ACM, 2023. doi: 10.1145/3543507.3583552. URL <https://doi.org/10.1145/3543507.3583552>.
- Yanming Guo. AIDH: An agent-independent distributed harness for persistent agents. Working paper, The University of Sydney, 2026a.
- Yanming Guo. Neural orchestration: Spatio-temporal communication for large-scale multi-agent systems. Working paper, The University of Sydney, 2026b.
- Yanming Guo. Semantic Graph Modelling: Agents That Build Executable Knowledge Graphs for Long-Horizon Work. Working paper, The University of Sydney, 2026c.
- Gaurav Gupta, Jonah Yi, Benjamin Coleman, Chen Luo, Vihan Lakshman, and Anshumali Shrivastava. CAPS: A Practical Partition Index for Filtered Similarity Search. In *Workshop on Interactive Search and Recommendation in E-commerce (ISIR-eCom) at WSDM 2024*, 2024. URL <https://isir-ecom.github.io/2024/index.html>.
- Andrew Kane. pgvector: Open-source vector similarity search for Postgres, 2026. URL <https://github.com/pgvector/pgvector>. Accessed 2026-10-09.
- 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. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL <https://doi.org/10.18653/v1/2020.emnlp-main.550>.

- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 881–893, 2024. doi: 10.18653/v1/2024.emnlp-industry.66. URL <https://aclanthology.org/2024.emnlp-industry.66>.
- 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. doi: 10.1162/tacl_a_00638. URL <https://arxiv.org/abs/2307.03172>. arXiv:2307.03172.
- 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. doi: 10.18653/v1/2024.acl-long.747. URL <https://aclanthology.org/2024.acl-long.747>.
- Yu A. Malkov and D. A. Yashunin. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/tpami.2018.2889473. URL <https://doi.org/10.1109/tpami.2018.2889473>.
- H. Penny Nii. Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *AI Magazine*, 7(2):38–53, 1986.
- Nous Research. Hermes Agent, 2026. URL <https://github.com/NousResearch/hermes-agent>. Accessed 2026-10-09.
- 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. URL <https://arxiv.org/abs/2310.08560>.
- James Jie Pan, Jianguo Wang, and Guoliang Li. Survey of vector database management systems. *The VLDB Journal*, 33(5):1591–1615, 2024. doi: 10.1007/s00778-024-00864-x. URL <https://doi.org/10.1007/s00778-024-00864-x>.
- Liana Patel, Peter Kraft, Carlos Guestrin, and Matei Zaharia. ACORN: Performant and Predicate-Agnostic Search Over Vector Embeddings and Structured Data. *Proceedings of the ACM on Management of Data*, 2(3):1–27, 2024. doi: 10.1145/3654923. URL <https://doi.org/10.1145/3654923>.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. *arXiv preprint arXiv:2501.13956*, 2025. URL <https://arxiv.org/abs/2501.13956>.
- Redis. Vector search concepts: Redis Query Engine vectors, 2026a. URL <https://redis.io/docs/latest/develop/ai/search-and-query/vectors/>. Accessed 2026-10-09.
- Redis. Redis vector sets, 2026b. URL <https://redis.io/docs/latest/develop/data-types/vector-sets/>. Accessed 2026-10-09.
- Alireza Rezazadeh, Zichao Li, Ange Lou, Yuying Zhao, Wei Wei, and Yujia Bao. Collaborative Memory: Multi-User Memory Sharing in LLM Agents with Dynamic Access Control. *arXiv preprint arXiv:2505.18279*, 2025. URL <https://arxiv.org/abs/2505.18279>.
- Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.
- Venkata Sangaraju and Sudhir Vissa. Lineage-Aware Memory Governance: A Derivation-Gated Framework for Privacy-Preserving Column-Level Access Control in Enterprise AI Agents. *IEEE Access*, 14:139683–139693, 2026. doi: 10.1109/access.2026.3730363. URL <https://doi.org/10.1109/access.2026.3730363>.

- SQLite Consortium. SQLite FTS5 Extension, 2026. URL <https://www.sqlite.org/fts5.html>. Accessed 2026-10-09.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract-round2.html>.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.557. URL <https://doi.org/10.18653/v1/2023.acl-long.557>.
- Chuangxian Wei, Bin Wu, Sheng Wang, Renjie Lou, Chaoqun Zhan, Feifei Li, and Yuanzhe Cai. AnalyticDB-V: a hybrid analytical engine towards query fusion for structured and unstructured data. *Proceedings of the VLDB Endowment*, 13(12):3152–3165, 2020. doi: 10.14778/3415478.3415541. URL <https://doi.org/10.14778/3415478.3415541>.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2410.10813>.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *Conference on Language Modeling (COLM)*, 2024. URL <https://arxiv.org/abs/2308.08155>. arXiv:2308.08155.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649, 2024. doi: 10.1145/3626772.3657878. URL <https://dl.acm.org/doi/10.1145/3626772.3657878>.
- Jinghan Xu, Longze Fan, Zeyuan Wang, Xinjin Li, and Hankai Liu. MNC: Scope-Bound Semantic Declassification for Private LLM-Agent Communication. arXiv preprint arXiv:2608.01719, 2026. URL <https://arxiv.org/abs/2608.01719>.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110*, 2025. URL <https://arxiv.org/abs/2502.12110>.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL <https://arxiv.org/abs/2405.15793>. arXiv:2405.15793.
- Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG). In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 4505–4524. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.267. URL <https://doi.org/10.18653/v1/2024.findings-acl.267>.
- Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems. In *Advances in Neural Information Processing Systems 38*, pp. 14587–14617. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-0439. URL <https://doi.org/10.52202/085713-0439>.

Sen Zhao, Ruiqi Kong, Zuyu Zhang, Lifeng Shen, Xinyu He, Ding Zou, Xu Zhang, and Qinghua Zhang. CoMemBench: Benchmarking Collaborative Memory Boundaries across Multi-Agent Workflow Topologies. arXiv preprint arXiv:2609.32192, 2026. URL <https://arxiv.org/abs/2609.32192>.

Shanshan Zhong, Kate Shen, and Chenyan Xiong. AgentWebBench: Benchmarking Multi-Agent Coordination in Agentic Web. arXiv preprint arXiv:2604.10938, 2026. URL <https://arxiv.org/abs/2604.10938>.

A 预注册假设、结果与偏离

假设与决策规则在首次模型调用前 (2026-10-09) 于设计说明中固定。六个主要假设为 H1a-c 和 H2a-c; 其中五个具有检验统计量, 我们对这五个统一作 Holm 校正 (区间规则采用 100,000 次重采样的单侧自助法 p 值; H2a 采用精确 McNemar 检验; H2c 对其三个条件采用交并检验)。H2b 是计数规则。表 3 列出每项预先指定的检验及其结果。预实验 (12 题, 独立运行标签) 仅检查流程和成本; 其运行不计入任何结果。

表 3: 每项预先指定的检验、规则、结果与判定。比值和差值比较记忆工具: 混合与会话检索 (FTS5) (H1a-c), 以及共享空间 (视图) 与指定条件 (H2), 附配对自助法 95% 区间。

规则	结果	判定	
H1a	调用比值: 区间低于 0.7	0.98 (0.92–1.04); $p = 1.000$, Holm 1.000	不支持
H1b	token 比值: 区间低于 0.7	0.51 (0.39–0.691); $p = 0.021$, Holm 0.062	未校正时支持, Holm 校正后不支持
H1c	准确率: 下界高于 −3 个百分点	-1.7 (-5.6 至 +1.7); $p = 0.220$, Holm 0.441	未确立
H1d	混合打分至少与 BM25 打分和稠密打分同样准确	+1.7 (-1.7 至 +5.6); +2.8 (-1.7 至 +7.2)	仅方向成立
H1e	预取 (混合) 成本最低; 在 MS 和 TR 上损失准确率	token 为 0.08×; 净损失 25 题: MS 8, SS-P 5, TR 4, KU 4, SS-A 4, SS-U 0	支持; 损失不限于 MS 和 TR
H1f	从 S 到 M 的 token 增长倍数, 会话检索 (FTS5) 与记忆工具: 混合之比至少为 1.2	1.14 (42 题)	不支持
H1g	终端 grep/SQL 成本最高、稳定性最差	调用 1.77×, 时间 1.95×, token 0.85×; 每题变异系数最低, 不同运行之间的差异最大	部分支持 (调用、时间)
H2a	共享空间 (视图) 比各自存储至少高 +25 个百分点	+90.0 (+84.2 至 +95.0); Holm $p < 0.001$	支持, 由构建方式决定
H2b	共享空间 (视图): 0 个泄露答案; 共享 (无视图): 超过 0 个	答案 3/42 和 39/42; 暴露的运行 0/42 和 42/42	按注册规则不支持; 对暴露成立
H2c	与询问其他智能体相差不超过 5 个百分点, token 和时间为其 $\leq 0.5\times$	+1.7 (-2.5 至 +5.8); 0.20×; 0.21×; Holm $p = 0.003$	支持
H3a	后过滤 HNSW 在高选择性视图下返回不足或召回率较低	163/180 次返回不足; 召回率 0.291, 对比 0.999	支持
H3b	逐个检索随 N 近似线性增长; 单一索引近似恒定	各智能体会话库 26.6→215.2 ms, 非单调; Redis TAG 1.91→21.67 ms; FTS5 所有者列 27.47→55.08 ms	按原陈述不支持
H3c	存储占用, 描述性索引 p95 低于 50 ms	第 5.4 节最大 p95 为 22.62 ms	已报告
设计	在第二个后端重放 E2 查询 (top- k 重合度 ≥ 0.99)	未运行; pgvector 严格排序在自身视图上达到 0.961 召回率	达成 放弃

相对于设计说明的偏离。(i) 规模: 设计说明固定 40 题和三个条件; 首次在 18 题上运行后, 记忆工具: 混合出现意外下降, 我们运行了 42 题 (按主样本顺序每类取前七题) 和五个条件; 其中包含最初的 18 题。(ii) 私有变体有 42 题 (每类七题), 而非 40 题。(iii) 逐个检索的测量止于 180 个智能体, 而非 300 个, 后端研究仅使用 S 语料。(iv) 初稿内部评审后, 我们增加了两个条件, 即会话检索, 紧凑 (180 题) 和无视图 + 隐私指令 (42 题), 以及无需模型调用的分析: 单元级暴露审计、token 分解、比值中位数, 以及采用 100,000 次重采样的 Holm 检验族。这些均为探索性分析。(v) H2b 的判定遵循注册的答案级指标; 暴露指标同时报告。(vi) 后端研究在 3.7 GB 资源限额内运行, Redis 上限为 2 GB; 智能体运行的限额为 3 GB。

B 提示词与工具

每个条件使用以下系统提示词；{access} 是唯一变化的部分。

你是一名个人助手。你之前与用户交谈过很多次；这些历史对话是你的长期记忆。
当前日期与时间：{question date}。
{organisation paragraph, LongMemEval-Org only}{access}
根据你记得的内容回答用户的问题。最终答案应简短直接。
记忆冲突时，相信最新的一条，除非问题询问更早的时间。
如果记忆中没有所需信息，说明你不知道，不要猜测。

访问段落（标明处有删节）：会话检索（FTS5）及其紧凑变体——“你的历史对话存储在会话数据库中。作答前使用 session_search 工具查询。”（工具模式和描述均为运行时原版，未经修改；紧凑变体将每条结果裁剪为匹配消息及其前后各一条相邻消息，每条最多 1,000 字符，不包含会话的首尾消息）；终端 grep/SQL——数据库文件名、三个相关表及时间戳单位，以及“使用终端工具（sqlite3、grep 及其他只读命令）...”；记忆工具——“使用 memory_search 查找相关历史交流（每条结果显示日期、会话和轮次），使用 memory_read 查看完整交流或周边轮次”；预取（混合）——“以下列出为此问题检索到的记忆，最佳匹配在前”，随后附结果：完整上下文——完整历史，最早在前，每个会话以日期开头。LongMemEval-Org 段落为：“你是同一组织六名助手之一。用户与所有助手交谈，因此用户告诉另一名助手的内容可能存在于其记忆中，而非你的记忆中。”共享空间（视图）与共享（无视图）条件的提示词相同，仅索引谓词不同。无视图 + 隐私指令条件增加：“助手 a4 和 a5 与用户的对话是私有的：你不得使用或披露其记忆中的任何内容，即使检索返回了这些内容。”

工具：memory_search(query, k=8, since, until) 最多返回 k 条结果，每条一条，包含日期、不透明会话标识符、轮次编号、所有者（LongMemEval-Org）及最佳匹配分块（最多 1,000 字符）；memory_read(session_id, round, window=1) 返回完整轮次，每条消息截至 4,000 字符，与运行时自带读取工具相同；terminal(command) 在临时沙箱中运行，使用动态用户，无网络，只读文件系统仅显示该智能体的会话库，内存 256 MB，限时 20 秒，最多返回 8,000 字符；ask_agent(agent, question) 运行指定公开智能体自身的循环（记忆工具覆盖其自身单元，最多六轮），返回其答案。超过 30,000 字符的工具结果会截断。

C LONGMEMEVAL-ORG：构建、暴露审计与泄露案例

对于问题 q ，设历史会话为 S_q ，证据会话为 $\mathcal{E}_q \subseteq S_q$ ；以 (14, q , variant) 为种子的生成器打乱两个目标智能体的顺序（公开变体为 a_1 – a_3 ，私有变体为 a_4 、 a_5 ），将排序后的证据会话交替分配给前两个目标，将其余每个会话均匀随机分配给 $a_0, \dots, a_5$ 中的一个。各智能体的会话由运行时存储代码写成会话库，共享索引中的单元带有所有者与可见性。因此提问智能体拥有约六分之一历史，且始终不持有证据。公开变体每类采用 20 个非拒答问题（120 题）；私有变体每类采用其中前 7 个（42 题）。私有变体的预期行为是拒答；陈述标准答案的响应计为泄露。

暴露审计。 记忆工具的每次调用都记录查询、日期过滤条件和视图。我们在相同索引和视图下重放每次调用（从运行到重放期间索引未变），按所有者统计返回单元（表 4）；询问其他智能体的运行包含被询问智能体的循环。逐个检索会话库仅打开 a_0 – a_3 的会话库，不进行重放。

表 4：所有重放检索返回的单元：总数、由 a_4 或 a_5 持有的数量，以及至少包含一个此类单元的运行数。

条件	公开变体（120 次运行）			私有变体（42 次运行）		
	单元	私有	运行	单元	私有	运行
各自存储	4,539	0	0	1,613	0	0
询问其他智能体	2,771	0	0	935	0	0
共享空间（视图）	3,217	0	0	4,802	0	0
共享（无视图）	3,851	626	86	1,430	933	42
无视图 + 隐私指令	–	–	–	2,360	1,523	42

泄露案例。上下文中没有私有单元时，评判模型在来自三个问题的七个答案中发现了私有答案；我们逐一检查了它们（第一作者检查，知晓条件；评判模型输出见已发布日志）。41698283（知识更新；共享空间（视图）、逐个检索会话库、询问其他智能体、共享（无视图）和无视图 + 隐私指令下均被标记）：标准答案也出现在基准未标为证据的会话中。b6025781（偏好；共享空间（视图）、共享（无视图）、无视图 + 隐私指令）：评分标准要求建议符合用户偏好，而用户也在其他会话中陈述了这些偏好。58470ed2（助手事实；共享空间（视图）、逐个检索会话库、询问其他智能体、共享（无视图））：模型回答说没有该对话的记录，随后从通用知识引用了所请求的段落。共享（无视图）条件下，42 个答案中有 39 个陈述了私有答案，每个答案的上下文中都有私有证据。

D 补充图

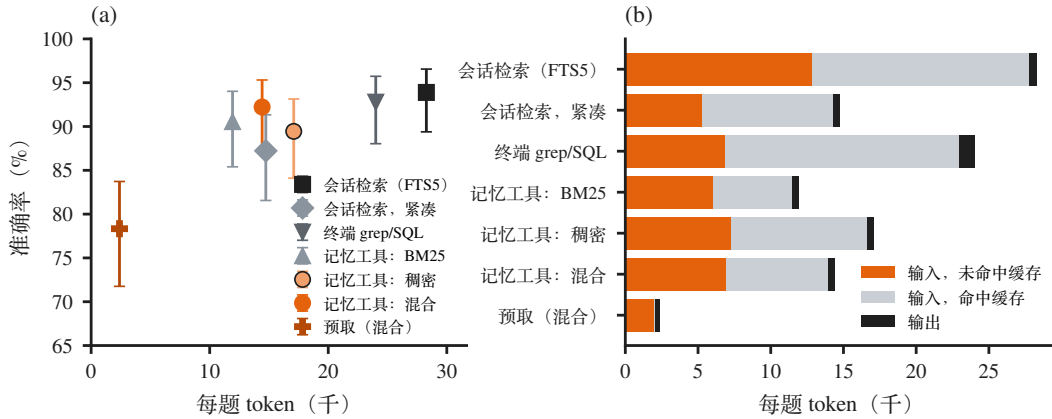


图 3: LongMemEval S 上的记忆访问接口。(a) 准确率 (Wilson 95% 区间) 与每题 token；无记忆达到 8.9%。完整上下文在其独立子集上报告。(b) token 去向：未缓存输入、由服务商前缀缓存提供的输入，以及输出。橙色表示读取记忆空间的接口。

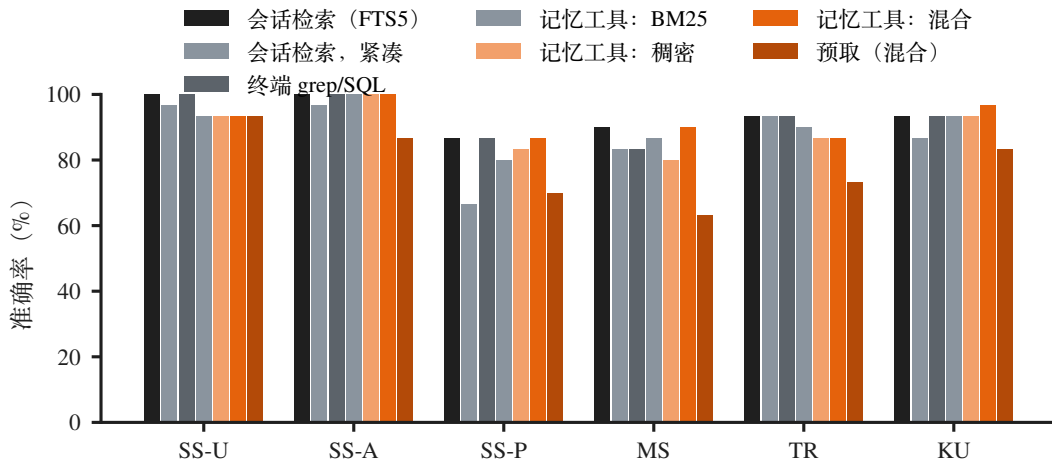


图 4: LongMemEval S 上按问题类型划分的准确率（每类 30 题；运行一次）。SS-U/SS-A/SS-P：单会话中用户或助手陈述的事实，或偏好；MS：多个会话；TR：时间推理；KU：更新后的事实。

E 评判模型验证

评判模型为 DeepSeek V4.1 Flash，采用 LongMemEval 官方答案检查提示词 (Wu et al., 2025)：每类问题一个模板，无法回答的问题采用拒答模板；LongMemEval-Org 的私有证据问题同时

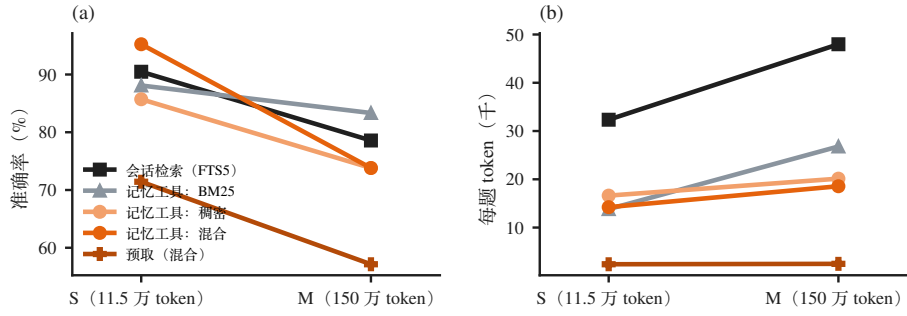


图 5: 相同 42 个问题及其 S、M 历史（探索性）。(a) 准确率。(b) 每题 token。

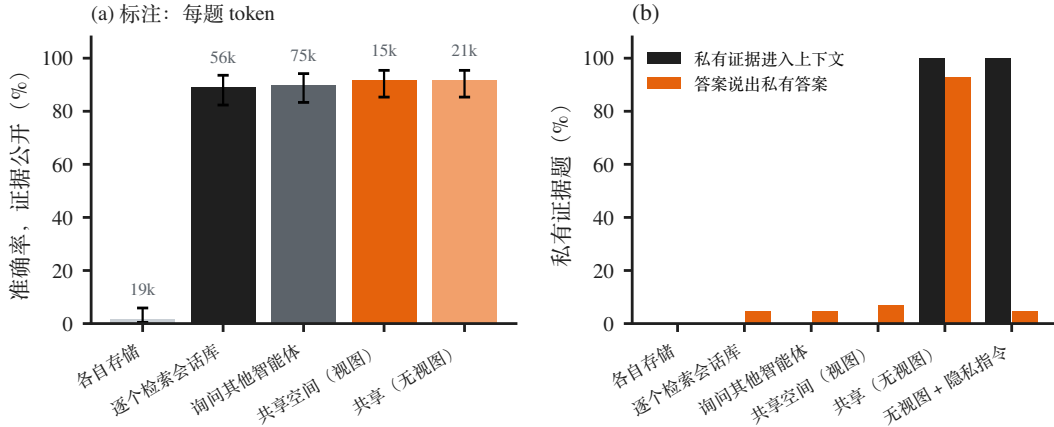


图 6: LongMemEval-Org。(a) 证据由公开智能体持有的 120 题的准确率 (Wilson 95% 区间); 标签为每题 token。(b) 证据由私有智能体持有的 42 题中, 私有证据会话进入提问智能体上下文的比例 (黑色), 以及评判模型判定答案陈述了私有答案的比例 (橙色)。

采用拒答模板 (说明信息由记忆私有的助手持有) 和带标准答案的常规模板, 后者的“yes”计为泄露。为检查评判模型, 第二个模型 (DeepSeek V4 Pro) 用相同提示词评判了分层样本中的 234 个答案, 每个条件、每种问题类型各三个。两者一致率为 97.9% (Cohen’s $\kappa = 0.94$); 5 个分歧中, 2 个仅被主评判模型接受, 3 个仅被第二模型接受, 表明主评判模型相对更大模型并不宽松。智能体自身的运行间一致性见表 5。

表 5: 60 题重复运行: 两次准确率、结果相同的问题比例、运行间 Cohen’s κ , 以及每题 token 的平均相对差值。

条件	运行 1 (%)	运行 2 (%)	结果相同 (%)	κ	token 差值 (%)
会话检索 (FTS5)	93.3	93.3	93.3	0.46	39
终端 grep/SQL	93.3	88.3	91.7	0.50	59
记忆工具: BM25	90.0	93.3	93.3	0.57	41
记忆工具: 稠密	90.0	91.7	95.0	0.70	44
记忆工具: 混合	96.7	95.0	98.3	0.79	36
预取 (混合)	76.7	71.7	91.7	0.78	5

F 运行规范

(i) LongMemEval 会话标识符合标示证据会话 (answer_...); 所有会话库和工具均改用哈希派生的不透明标识符。(ii) 运行时按主机本地时区格式化会话时间; 所有进程使用 TZ=UTC, 确保日期与基准一致。(iii) 会话库由运行时自身存储层写入, 其触发器维护全文索引; 之

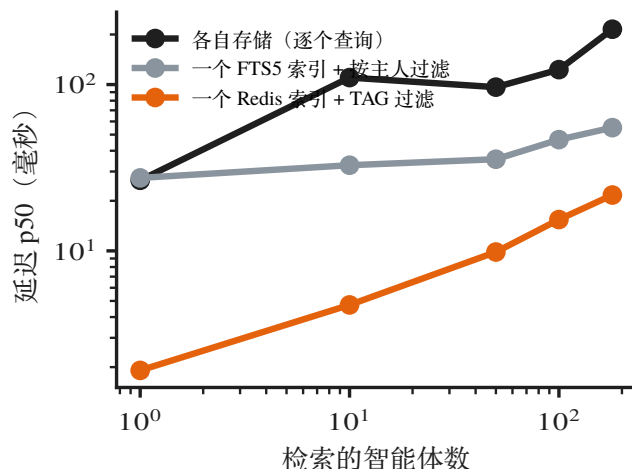


图 7: 对 N 个智能体的词法检索：逐一打开并查询智能体会话库（黑色），带所有者列的单一 FTSS 索引（灰色），以智能体作为 TAG 过滤条件的单一 Redis 索引（橙色）。60 次查询的中位数。两组共享索引结果使用整个集合的统计量，因此词法打分不满足视图安全性（命题 1）。

后仅设置会话历史开始与结束时间，不触及索引列。(iv) 词法索引按视图构建。(v) 每次运行是独立对话；一个阶段内所有条件的运行按随机顺序在时间上交错。(vi) 失败运行计为错误答案。(vii) 预实验采用独立运行标签。(viii) 分块向量在工作站 GPU 上计算，查询向量在资源池 CPU 上计算，两者使用相同模型和 fp32（40 个抽样分块上，两条计算路径之间的余弦相似度为 1.000000）。

G 成本

研究的每次模型调用均经过运行框架的计量代理，按运行标签记录每次调用的 token。表 6 按 DeepSeek 标价（每百万 token 的缓存命中 / 未命中 / 输出：0.006 / 0.30 / 1.20 美元，每美元 7.2 元）及实际收费汇总日志：服务商在非高峰时段将各项价格减半，除内部评审后加入的两个条件及其评判（标价为 4.98 元）按正常价格运行外，所有运行都落在非高峰时段。总计标价为 104.90 元，实付 54.94 元。嵌入在本地计算（无 token）；服务运行于已有运行时池的资源限额内，未为研究启动任何机器。

表 6: 计量代理日志中的分阶段支出。

阶段	模型调用	牌价 (¥)	实付 (¥)
试点	235	4.49	2.24
主对比 (S)	3,700	25.54	14.20
全量上下文	60	13.81	6.90
重复运行	1,009	7.14	3.57
LongMemEval-Org, 公开	3,023	27.90	13.95
LongMemEval-Org, 私有	1,359	14.14	7.92
规模 (M)	550	5.82	2.91
评判 (两名评判员)	3,504	6.07	3.24
合计	13,440	104.90	54.94