认知介导的记忆生命周期（CMML）：长期运行智能体记忆失效的阶段分解与判定

——一个长期运行个人智能体的观察研究

The Cognitively Mediated Memory Lifecycle (CMML): Stage Decomposition and Decision Rules for Memory Failure in Long-Running LLM Agents

作者：Nether_Dream ｜ ORCID：0009-0002-5801-9655 ｜ 单位：独立研究者（Independent Researcher） ｜ 通讯邮箱：Nether_Dream@outlook.com

日期：2026-10-31（版本日期；观察数据截止 2026-10-28；系统规模审计基线 2026-10-07） ｜ 文稿性质：预发行稿（概念与方法）

状态说明：本稿为预发行阶段的概念与方法稿，不含受控实验，也不报告性能提升。全部量化陈述均标出取值日期与口径（附录 H、I）；20 条参考文献经文献核验工具逐条比对（核验情况见参考文献末注）。文中标为「未测量」「未确认」「暂不可重跑」的环节是本文主动声明的边界，不是待补的遗漏。

摘要

基于大语言模型的智能体已经能够保存任务状态、写入记忆文件并跨会话继续工作，但保存并不保证使用。本文提出的认知介导的记忆生命周期（CMML）把从保存到影响结果的过程切成七个可分别观察与判定的阶段——存储、检索、呈现、评估、决策影响、行为有效与更新——并为每个阶段给出操作性定义、判定规则与典型失效模式。

这套分解的动机是一条可检验的命题：记录存在不蕴含记录会被想起，中间需要一个显式的触发与判定过程。已有以"记忆生命周期"为名的框架分别关心记忆以什么形态存续、记忆通路如何被攻击与治理，都不区分"在上下文里"与"改变了行为"之间的四步；CMML 补的正是这一段。

本文的交付物是一项主张与五项方法产物：在一个持续运行一年以上的个人智能体上给出各阶段的实测锚点；把观察到的失效整理为按位置与按内容两维共四类；以单条记忆为单位、用 15 个案例逐条给出会话标识、时间点、证据分级与判定状态；提出"复述率"与"提问次数"两个可直接采集的指标；以及在冻结核心模型条件下的对照实验设计。

本文属于预发行阶段的概念与方法稿，不报告受控实验，也不主张 CMML 已被证明能提升智能体性能。文中标为「未测量」「未确认」「暂不可重跑」的环节，是作者主动声明的边界，不是待补的遗漏。

Abstract

Large language model (LLM) agents can already persist task state, write memory files, and resume across sessions, yet persistence does not guarantee use. We present the Cognitively Mediated Memory Lifecycle (CMML), which decomposes the path from storage to influence into seven separately observable and adjudicable stages -- Stored, Retrieved, Presented, Evaluated, Decision-influential, Behaviorally effective, and Updated -- and gives each an operational definition, a decision rule, and a characteristic failure mode.

The decomposition is motivated by one testable claim: that a record exists does not entail that it will be recalled, so an explicit trigger and adjudication step must sit in between. Existing frameworks that use the phrase "memory lifecycle" ask instead what form a memory should persist in, or how the memory pathway can be attacked and governed; none separates the four steps between "in the context" and "changed the behaviour". CMML addresses that segment.

The paper delivers one claim and five methodological products: measured anchors for each stage on a personal agent that has run continuously for more than a year; a two-axis taxonomy of observed failures; a per-memory diagnostic method with 15 cases, each recording session identifiers, timestamps, an evidence grade, and a adjudication status; two directly collectable metrics (restatement rate and question count); and a controlled-experiment design under a frozen core model.

This is a conceptual and methodological preprint. It reports no controlled experiments and does not claim that CMML improves agent performance. Items marked unmeasured, unconfirmed, or not-yet-reproducible are declared boundaries of the work rather than unfinished obligations.

1. Introduction　引言

本章给出问题、机制命题与研究问题。第 2 章定位相关工作；第 3 章给出 CMML 的七阶段定义与判定规则；第 4 章给出可观测证据与失效分类；第 5 章给出评估设计；第 6 章讨论边界与局限；第 7 章结论。运行方法细节、相关工作展开与数据口径收在附录。

1.1 Research Background　研究背景

基于大语言模型的智能体能够通过模型生成计划、调用工具并根据环境反馈继续行动。相关研究已经展示了多条路径：ReAct 将推理轨迹与行动交替组织，使模型能够根据观察结果继续推理与行动[1]；Reflexion 使用语言形式的反馈与情节记忆，在不直接更新模型参数的前提下改善后续尝试[2]；Voyager 通过不断扩展的可执行技能库、自动课程与环境反馈支持开放式探索[3]；Generative Agents 以自然语言保存完整经历，并通过反思与动态检索支持行为规划[4]。这些工作共同说明，智能体的行为不只取决于基础模型参数，也受到上下文、工具、记忆、技能与反馈组织方式的影响。

长时程运行带来一个更细致的问题：保存信息并不等于在需要时使用信息。一条经验可能已经写入文件或数据库，却没有在相关任务出现时被检索；检索结果也可能已经进入上下文，但未被核心模型采纳；模型即使引用了历史信息，也未必因此改变判断或行动。这三个断点（未检索、未采纳、未改变行为）在现有评估中常被合并为单一指标，从而无法定位失效发生在何处。因此，评估智能体记忆不能只看存储容量或检索命中率，还需要区分记忆的可用性、上下文呈现、模型评估、决策影响与行为结果。

1.2 Problem Statement　问题陈述

本文归纳一条可检验的机制命题，以及一条尚待检验的假设。前者断在存储与检索之间，后者断在呈现之后。

核心命题：智能体读取自己的工作区记录时，记录本身不会像人类回忆那样把智能体拉回事件现场；因此"记录存在"不蕴含"记录会被想起"，中间需要一个显式的触发与判定过程。

待检验的假设：呈现内容的变化，可能比呈现的次数更影响它随后是否被处理。依据来自注意力的常见现象：反复出现的相同内容不易再引起注意，而变化会重新引起注意。

这条假设目前没有已确认的支撑案例。曾经被当作其证据的一条，在核对日志后改判为「完成未回写」：动作早已完成，缺的是状态回写。重复是任务列表里那条状态没更新的征兆，不是被忽略的原因；从"重复且无进展"推到"重复导致忽略"，是一次证据跳跃。

另有一项实现层的事实只说明一半：被研究系统的抽样作业带有「内容不变即压制」的闸门，闸门输出与被跟踪内容逐字节相同时整轮不调用核心模型。这证明"零变化的通道不会唤醒模型"，但不证明"变化就能提高处理概率"。

这条假设以 F4 列在 1.3 节的未来研究问题中。

该命题的直觉是：人类看到一行旧笔记，往往会被线索带回当时的场景、目的与后果；而对语言模型而言，一段被写入文件的文字在未被读入上下文之前，与不存在没有区别；即便被读入，它也只是当前上下文中的一段普通文本，不与当时的行动、失败或代价绑定。这解释了三个设计后果：需要对“什么内容常驻眼前”设置预算而不是全量注入；需要低成本线索（关键词、引子）而不是让模型凭记忆想到去查；以及需要区分"被检索到"与"被采纳"这两个在人类回忆中往往合并的环节。

本文的问题意识与全部运行事实来自 Lilith（莉莉丝）：一个基于 Hermes 运行时的长期运行个人智能体，主模型为云端 API，本机另驻四个小模型承担视觉、批量阅读与向量检索等专项工作。在本文中她属于研究对象。其可核查的基本规模（下引数值为 2026-10-07 审计值；2026-10-20 复测值见附录 H）为：会话库 322 个会话 / 217,552 条消息（state.db 约 1.3 GB）、技能 193

个、框架 cron 任务 22 个、Windows 计划任务 13 个、记忆文件由 git 仓库管理并已有数百次带说明的提交 [运行][源码]。

在这一系统上，研究者记录到四类现象，并有可复核的证据：

· 现象 A（未检索）：一条「需要检索时应使用浏览器」的规则写在常驻记忆文件第 8 行，需要时没有浮现；同期另有两次同类事件被系统自身归为同一模式 [运行]。

· 现象 B（索引不覆盖）：语义索引只收集记忆与技能目录，文档目录的收集语句缺失，613 个被索引文件中该目录 0 条 [源码][运行]。

· 现象 C（呈现有预算）：常驻注入层有硬性字符上限并曾触发裁剪，某常驻记忆文件在 2026-10-02 的十余分钟内由 15,079 字节降至 10,135 字节，此后在 10.1–11.9 KB 间波动 [运行]。

· 现象 D（完成而未回写）：一条待办的验收在 2026-10-26 凌晨已完成，但"已完成"从未回写到条目，此后它以未完成身份被早间作业重述，21 天内 19 次、其中 11 次逐字相同 [运行]。

四类现象的完整证据与逐条判定见 4.2 节与附录 B。它们只是研究动机，不足以单独确定原因：索引覆盖、触发条件、上下文位置、指令竞争与指令遵循能力都可能是影响因素，本文不把其中任何单一因素视为已证实的因果解释。有一点需要强调：现象 B 属于实现缺陷，发生在任何模型被调用之前——记忆失效可以出现在完全非认知的层次上，把它与"模型没有采纳"混为一谈会导致归因错误。

1.3 Research Questions and Contributions　研究问题与贡献

本文的核心交付物是 CMML 这一套阶段分解与判定规则。本节把问题分三层：一个核心问题、三条已给出答案的设计主张、四个留待未来研究的问题。三者的证据状态不同，不应并列陈述：核心问题由本文回答，设计主张是配套立场，未来问题明确不在本文范围内。

核心问题（本文回答）

Q1（记忆生命周期）：如何描述记忆从形成、保存、检索、进入上下文，到可能影响决策与后续记忆的完整过程，并使其中各阶段可被观测、可被判定？

子问题 Q1.1：如何让记忆真正影响后续行为，而不只是停留在"被检索到"或"被写入"？

设计主张（本文已给出答案，不作为待答问题）

D1（适用性判断）：候选记忆是否适用于当前情境，应由核心模型判断；辅助模型只负责以低成本提出候选。

D2（双层记忆）：紧凑的事实记忆与会话原文记录是互补的两层资源，不能相互替代。

D3（运行时分层）：运行时的执行控制应与核心模型的认知判断分层；对具有外部副作用的动作，单独定义权限与恢复语义。

这三条是本文的设计立场，不是已验证的结论。它们的实现状态并不一致，逐项判定见 3.5 节。

未来研究问题

F1 工作空间演化：核心模型参数不变时，工作空间的组织与内容演化能否带来稳定、可重复的长期提升（子问题 F1.1–F1.5 见附录 G）。

F2 共同演化：核心模型与工作空间能否共同演化，边界在哪里。

F3 具身扩展：CMML 能否扩展到机器人等具身系统。

F4 呈现变化与后续处理：呈现内容的变化是否比呈现次数更影响它随后是否被处理。这是 1.2 节那条尚待检验的假设，目前没有已确认的支撑案例；检验需要在内容不变、内容变化、变化幅度不同三种条件下比较同一任务的处理率与交互率。

三层与章节的对应：Q1 由第 3 章回答；D1–D3 的职责划分在附录 D 展开；F1 的实验设计在第 5 章；F2 在附录 C。

候选贡献

本文的贡献是一项主张加五项方法产物。

主张：记录存在不蕴含记录会被想起（1.2 节）。该命题给出可被证伪的预测；由注意力现象引来的一条推论目前无可核实案例，列入 F4。

方法产物：

- 把既有的粗粒度层级区分细化为七个分析阶段，并在"取出之后"补出四个可分别判定的环节（呈现、评估、决策影响、行为有效），为每个环节提出操作性定义、候选观测事件与判定规则（3.2、3.5 节）；规则的独立标注一致性待验证。
- 在一个真实长期运行系统上给出各阶段的实测锚点（3.5 节）。
- 把观察到的失效整理为按位置（未写入、层错位）与按内容（完成未回写、内容与判断缺陷）两维共四类，并保留一个标注为"未确认"的形态（4.2 节）；其中「完成未回写」与「拒绝但改变计划仍算第 5 阶段成立」是本文的判断，不在既有分类之内。
- 给出一套以单条记忆为诊断单位的判定方法：15 个案例逐条给出会话 id、时间点、证据分级与判定状态，含 1 个标注"不采信"的反例与 3 次改判记录；并在 4.5 与附录 I 给出用推理轨迹回溯第 4 / 5 阶段的流程（覆盖率 97.8%）。

· 提出「复述率」与「提问次数」两个可直接采集的指标，以及冻结核心模型的对照实验设计、主次指标与噪声控制方法（4.5、5.2 节）。

边界说明：本文不宣称 CMML 是相关领域的首创。Agent 运行时、Agent 记忆与长期运行智能体都已有明确的先行工作（附录 E.2），把"多模型加运行时"或"给智能体加记忆"当作新颖性本身是不成立的。本文的主张集中在三点上：阶段分解的可判定性、阶段实测锚点，以及冻结核心条件下的实验设计。

1.4 Paper Organization　论文结构

正文保留主张、证据与判据：第 2 章只给比较结论，第 3 章是主体（七阶段定义、可观测事件、判定规则与实测锚点），第 4 章给可观测证据与失效分类，第 5 章给评估设计，第 6 章讨论边界与局限，第 7 章结论。其余材料——术语、观察案例集、工作空间演化、运行方法、相关工作、图表清单、研究问题、数据口径——收在附录 A 至 I。

2. Related Work　相关研究

本文与三条技术线有关：Agent 运行时、Agent 记忆与长期运行智能体。它们与本文的逐项比较、阶段对照以及各文献的核验层级收在附录 E；这里只保留比较所需的结论。

三条线共享一个前提：长期运行智能体的问题不能只在模型层解决。本文与它们的差别不在机制，而在研究对象——它们多以运行时机制或记忆策略本身为贡献，通常需要在受控环境中构造实验以证明机制收益；本文则在一个已经持续运行的个人系统上，问"运行时提供的记忆资源有没有真的进入认知过程"，并把运行时的角色限定为事实与权限的提供者（附录 D）。

逐行比较见附录 E 的表 E-1（15 项工作与本文的机制对照），各行的核验层级见表 E-2。

3. CMML: Cognitively Mediated Memory Lifecycle　认知介导的记忆生命周期

本章是本文的主体。它给出七阶段的定义、可观测事件、判定规则与实测锚点。

3.1 Definition and Scope　定义与范围

本章是全文的核心。3.1 节给出主张与两个过程的区分；3.2 节给出七阶段与概念流程；3.5 节给出各阶段的可观测事件、判定规则与实测锚点；3.4 节讨论记忆类型与存续生命周期；3.6 节论证相邻阶段为什么不能合并。

Cognitively Mediated Memory Lifecycle（CMML，认知介导的记忆生命周期）是本文用于描述记忆与当前认知过程关系的方案名称。这里的「周期」是字面意思：第 7 阶段（更新）产生或修正的条目会回到第 1 阶段（存储），因此它是一条闭环，不是一条有终点的单程。单个条目可能停在任何一环——多数在前三段就终止——这不改变结构本身是循环的：条目一旦被更新，就会回到起点再走一遍。其核心关注点不是单纯降低无关上下文，而是：记忆作为外部信息资源，如何在适当时机进入当前认知过程，并可能影响模型的解释、判断、计划与行动。

「介导」指的是哪一层。 记忆不能直接作用于行为：它必须先经过一次显式的评估（第 4 阶段），才可能影响判断与行动。这道评估就是介导层。本文之所以把它单列为一个阶段，是因为它最容易失效，而失效又最不容易被看见。

关于术语来源的说明。 CMML 借用记忆科学的词汇（生命周期、介导、再巩固），但七阶段的划分来自智能体运行时上可观测的环节，不是对人脑记忆机制的建模。就认知心理学而言，与第 4、5 阶段最接近的是元记忆的监测—控制框架[15]：第 4 阶段对应监测（这条可信吗、适用于当前情境吗），第 5 阶段对应控制（据此改变或不改变行动）。这一类比有独立证据支撑——在 20 个前沿模型上，回答准确率的排名与元认知敏感度的排名「largely inverted」[16]，即「答得对」与「知道自己答得对不对」是两件不同的事，这正是不把评估与行为有效合并的理由。第 7 阶段回到第 1 阶段的回环则接近「再巩固」：提取会令记忆痕迹重新进入可修改状态[17]。

这一层是借词，不是依据：本文没有从上述工作推出七阶段。

记忆的「动态性」不意味着每条记录都必须持续改写。这里需要区分两个过程，它们容易被混为一谈：

其一，认知参与——过去的记忆进入当前判断并可能影响决策，这个过程不需要改写记忆本身；其二，生命周期更新——当前判断的结果、实际行动或反馈反过来影响记忆与经验，这个过程可能需要改写。两者可以相连，但不是同一件事。因此，一条记忆即使保持原样，只要它能在后续事件中再次参与判断，就已经不再是静态档案；反过来说，一条记忆被改写了，也不代表它真的参与过判断。

本文的可检验性主张是：七阶段可以被分别观测，且阶段之间存在可测量的衰减；若该衰减在受控条件下不可复现，则 CMML 的实用价值应当被重新评估。

3.2 Logical Structure　逻辑结构

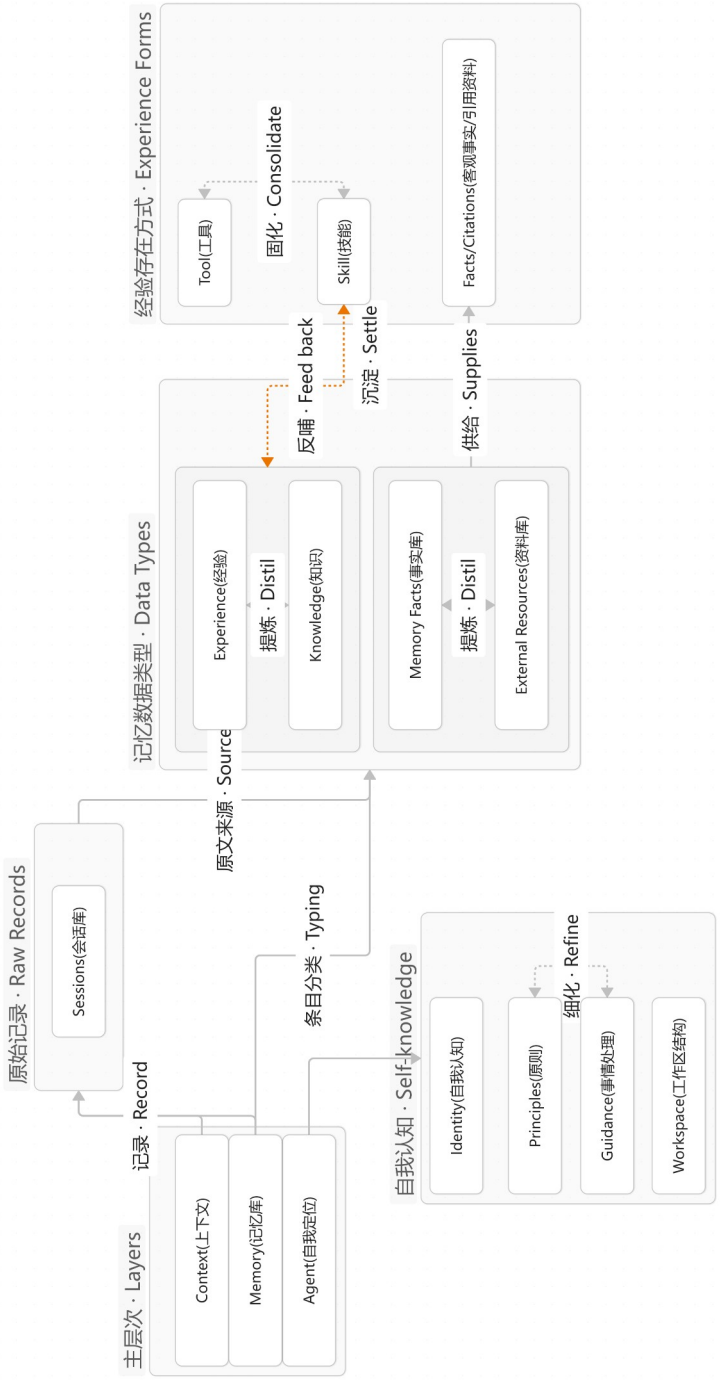

图 1　CMML 逻辑架构：Context、Memory、Agent（自我定位）与自我认知、Sessions、记忆数据类型与经验存在方式

Figure 1. CMML logical architecture: Context, Memory, Agent, Sessions, memory data types, and modes of experience

图 1 表达的是层次关系，不是执行次序：Context 是当前这一轮可见的内容，Memory 是持久存储、按需检索，Sessions 是原始记录，Agent 是读取它们的那一侧。检索可以由运行时、辅助模型、规则或核心模型发起，具体触发条件、召回次数、排序方法、冲突处理与写入策略需要通过工程设计与实验确定；各阶段的判定规则见 3.5 节，这是使该结构可被独立复现的关键。

3.3 Runtime Process　运行流程

从输入到存储的运行流程可概括为：用户输入与该轮上下文进入处理流程；低成本辅助模型提供关键词或候选线索；核心模型判断是否需要回忆；需要时由检索环节执行候选召回；结果返回核心模型；核心模型继续处理事件。上下文长度不在反馈路径上：当前 token 量由插件或核心模型侧读取，达到阈值时由核心模型决定是否压缩，并审阅压缩后的内容，确认后写入存储层。

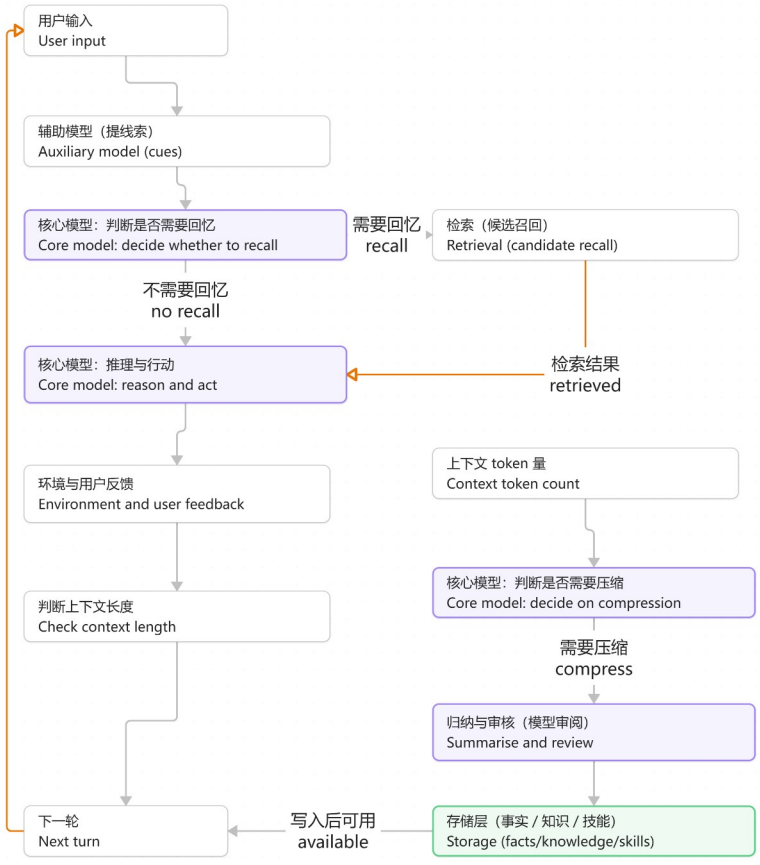

图 2　CMML 运行架构：检索回环、轮次回环，以及与反馈无关的压缩支线

Figure 2. CMML runtime architecture: the retrieval loop, the turn loop, and a compression branch independent of feedback

3.4 Memory Types and Lifecycle　记忆类型与生命周期

事实记忆与会话记录是互补而非相互替代的两层资源。事实记忆提供紧凑入口，便于快速查找稳定事实或摘要；会话记录保留信息形成时的语境、用户修正、任务步骤与结果，可在摘要不充分、信息冲突或需要追溯来源时提供依据。

该设计不预设具体的自动回溯触发阈值。可作为回溯候选条件的包括：事实记忆置信度不足、来源之间冲突、适用条件不明确或缺失、任务风险较高、以及第 4 阶段判定为「不适用」但任务仍需要该主题信息。是否应在这些条件下回溯源会话，以及其对判断质量、延迟与上下文成本的影响，属于第 5 章的实验内容，本文不给出结论。

成功与失败都可以成为经验材料。成功经验可能记录有效策略、适用条件与可复用步骤；失败经验可能揭示无效路径、工具限制、环境条件或错误假设。失败不应仅因结果不佳就被全部丢弃。

保存失败不等于把失败过程中的所有判断当作真理。失败可能来自模型推理错误、工具故障、信息不足、环境变化或目标定义不清。经验记录应尽可能区分：任务目标、当时条件、采用的行动、可观察结果、失败原因（若已确认）、不确定性与后续修正。历史结论应附带适用条件，避免在相似但不同的情境中被机械复用。为此，本文建议经验条目采用如下最小字段集。

表 2　经验条目的建议最小字段集

字段	用途	缺失时的后果
任务目标	界定该经验适用于哪类任务	结论被跨任务套用
当时条件（环境、工具版本、模型版本）	界定适用前提	在条件变化后仍被复用
采用的行动序列	提供可复用或可规避的路径	只能记住结论，无法复现过程
可观察结果	区分推断与事实	归因建立在推测之上
失败/成功原因（若已确认）	支撑因果性说明	错误归因被长期保留
不确定性标记	提示模型谨慎使用	低置信度结论被当作事实
后续修正记录	支持冲突解决与更新	冲突条目长期并存

就存续而言，记忆的生命周期可以表述为：进入（Admission）→ 活跃（Active）→ 降级（Downgrade）→ 归档（Archive）→ 删除（Delete）。其中删除是最后才考虑的选项：对研究系统而言，历史证据、可回滚能力与因果追踪本身具有价值。

3.5 Decision Rules and Measured Anchors　判定规则与实测锚点

下表为七个阶段分别给出：可观测事件（运行时或日志中可直接记录的事实）、判定规则（何种证据足以判定该阶段成立）、以及典型失效模式。判定规则的设计原则是：只依赖可被第三方检查的记录，不依赖模型的自述（理由见 附录 D）。

表 3　CMML 七阶段的可观测事件、判定规则与典型失效模式

阶段	可观测事件	判定规则（成立条件）	典型失效模式
1 Stored	工作空间写入记录、写入时间、来源标识	存在可被独立读取的记录，且带有来源与时间戳	未写入；写入但无来源；写入内容与原始语境脱节
2 Retrieved	检索调用、查询串、候选集与排序分数	候选集非空，且查询由当前任务触发而非人工提醒	未触发检索；触发但候选为空；候选与当前任务主题错配
3 Presented	运行时注入记录：实际进入上下文的片段与 token 预算占用；该片段距上次呈现是否发生内容变化	该片段出现在本轮模型输入中（以注入日志为准，不以模型自述为准）；并须单独记录"本次呈现与上次相比是否有 delta"——无 delta 的重复呈现只满足可见性，不满足显著性	被排序截断；被去重或压缩规则移除；注入位置导致被忽略；内容零变化而退化为背景
4 Evaluated	模型输出中对候选记忆的显式取舍、适用性判断或质疑	输出中存在可定位的取舍表述，或存在因该候选而改变的计划要素	未作判断直接采用；未作判断直接忽略；误判适用性（把不适用当作适用）
5 Decision-influential	推理链、计划或工具调用参数中可追溯地使用了该记忆内容；或该记忆被明确评估后拒绝采用	该记忆影响了下游可观察产物——包括以「不采用」的形式影响：模型改变了原计划、避开了历史失败路径，或明确给出不适用理由	口头引用但下游无变化；引用后仍按原路径执行；未作评估即忽略（无法与未检索区分）
6 Behaviorally effective	行动序列、工具调用结果、任务终态	在同一任务的对照条件下，采纳该记忆与否导致行动或结果发生差异	采纳但被工具失败抵消；采纳但被其他上下文指令覆盖
7 Updated	工作空间写入 / 修改记录、修订原因、被修正的对象	产生新的或修正后的记忆条目，且内容发生语义变化（新增、修正、删除、拆分、状态变更），且可追溯到触发它的任务结果；原样重写不计为有效更新	只追加不改写；原样复述（写了但没变）；错误归因被写入；冲突未解决而并存

术语说明： 第 5 阶段定名为 Decision-influential（决策影响）。该阶段的操作定义同时包含两种情况：接受记忆内容并使用（结论复用或推理迁移），以及在评估后拒绝该记忆、但拒绝本身改变了下游计划或行动。后者在日常语义中不符合“采纳”，因此不采用 Adopted 作为阶段名；若需引用旧称，可在首次出现处注明「曾称 Adopted」。

一条记忆可能同时在多个阶段失效。本文的规则是：判定取最早失效的那个阶段，其余阶段记为并列失效并在案例中分别列出——这样同一个案例不会因为没有唯一归属而被随意归类。

第 7 阶段还需要一条判据：一次写入究竟是有效更新，还是无意义的改写。本文采用三条并列条件——内容变化是否改变了后续的检索或决策路径；是否消除了既有的冲突或冗余；是否携带了可执行的适用条件（何时该用、何时不该用）。三条都不满足时，即使字数增加，也只记为一次改写事件，不记为更新。

这条判据也解释了一类容易被误读为「模型变聪明了」的现象。同一个同质事件第二次处理得更好，可能来自那次写入真的被后续任务用上了，也可能只是模型这次恰好找到了一条更好的路径；二者都会表现为「第二次更好」，要区分它们，需要同一任务的对照，而不是次数统计。反过来，有些改进只可能来自记忆——用户的习惯、口头约定、曾经明确否定过的做法，这些信息在模型自身能力里不存在，只能靠条目留存并在下次被取用。判断一次「变好」是否属于记忆效应，先看它依赖的信息在不在模型可见范围内。

关于判定规则有四点需要补充。第一，阶段的判定不能只依据模型对记忆的复述，因为模型可能生成形式上引用而实质上未改变下游行为的表述。第二，「Decision-influential」不等于「照搬历史结论」：一条记忆被检索、被评估，并导致模型避开历史失败路径或改换方案，仍然属于影响了下游产物；把「拒绝采用」排除在外，会系统性漏计真实的记忆作用。第三，因此判定需要证据组合，而不是单一信号：仅有复述不足为据；复述加下游变化足以判定被采纳；复述加明确的不可用性说明与方案改变，同样足以判定产生了影响。第四，「Behaviorally effective」本质上是反事实判断，只能在对照条件下估计，单一轨迹中无法确证（见 附录 D 与第 5 章）。

第五，「Presented」需要区分可见性与显著性。片段进入上下文只证明可见，不证明被处理。系统可以观测的代理量有两组：该片段距上次呈现是否发生内容变化（delta），以及该片段是否引发了一次向用户的提问。前者的含义见 4.2.3；后者是本文新提出的指标，二值、可数、可审计——一条长期挂起的事项若从未引发提问，它的"被呈现"就只是形式上的。

该阶段划分是本文用于澄清问题与设计实验的分析方案，并非声称所有智能体实现都以相同模块存在。近期面向长时程智能体记忆的评测工作也强调需要在任务环境中区分记忆能力与推理、检索、工具使用等因素[9]；关于记忆机制的综述则从写入、管理、读取等环节对既有系统作了系统梳理[18]。上述文献描述仍须以原文逐项复核，本文不把文献所采用的术语或评测目标等同于 CMML 的七阶段定义。本文与二者的差异在于：本文把阶段之间的衰减本身作为测量对象。

为使上述判定规则不停留在分类层面，下表给出七个阶段在被研究系统上的实际对应物与取值。这些取值的用途是证明“阶段可被观测“，而不是证明”某个机制更有效”。所有条目均给出可复核的证据类型（源码行、记录、快照或实测）。

表 4　CMML 七阶段在被研究系统上的实测锚点

阶段	系统上的实际对应物	取值 / 事件	证据
1 Stored	文件式长期记忆 + 事实库 + 向量索引；记忆目录由 git 管理	长期记忆顶层约 52.6 万字符（2026-10-07；2026-10-20 复测 67.9 万）；记忆仓库累积 591 次带说明的提交	[运行][源码]

阶段	系统上的实际对应物	取值 / 事件	证据
2 Retrieved	语义索引 + 关键词入口 + 引子层按需读取；检索由工具调用触发	索引 613 文件 / 12,407 块（构建于 2026-10-07，至 10-20 未再重建），文档目录 0 条（收集循环缺少收集语句）；2026-09-30 至 10-05 之间该检索工具共 16 条调用记录、10 个不同调用时刻	[源码][运行]
3 Presented	常驻注入层（有硬性字符上限）+ 按需注入	常驻注入约 1.95 万字符（2026-10-07，四个文件；2026-10-20 复测三个文件合计 1.60 万，SOUL 已不存在）；某常驻文件 2026-10-02 由 15,079 裁至 13,832 字节、同日再降至 10,135；固定背负约 93 KB/轮（2026-09-29 实测，含工具表），当前值未重测	[运行]
4 Evaluated	核心模型在检索入口处判断是否需要回忆；总结内容提交主模型审核，不通过则重新总结	该判断点在设计文档与运行架构中均被显式标出；审核—重做回路有文档定义	[源码][运行]
5 Decision-influential	记忆内容进入计划、工具参数或约束遵守；尚无独立的采纳记录	审计给出的四条“影响了行为”证据均为相关证据；系统自身声明无法证明因果	[自述][推断]
6 Behaviorally effective	任务终态与工具结果	未做有无记忆的对照，因此该阶段在当前系统上未被测量	[假设]
7 Updated	记忆文件的写入与修正；经验条目带回适用条件	存在一条待办在 21 天内被原样重述 19 次（其中 11 次逐字相同）、零次提问、零次状态变更的实例，说明「写入发生」不等于「更新有效」	[运行]

需要补一句边界：第 5 阶段目前只有案例级的相关证据，是否满足表 3 的每条操作判据需逐案复核；第 6 阶段没有任何已完成的行为有效性测量。本文不报告行为有效率，也不据现有案例推断记忆带来了性能提升。

这张表本身给出一个结论：七个阶段并非同等程度地可观测。存储、检索、呈现三阶段在当前系统上已有硬证据（计数、索引、预算）；评估阶段有明确的判断点但缺少日志；采纳与行为有效两阶段只有相关证据，因果未被测量。因此本文把第 5 至第 6 阶段的测量列为后续工作的第一优先项，而不是假定它们已经成立。

另一个可直接使用的量是写入与命中之间的延迟。本系统上有一组可复算的记录：一次注入层裁剪涉及 19 个条目，此后在检索记录的全部 411 条结果、29 个去重查询串中，没有任何一次指向这些条目。这一结果需要谨慎表述——它不证明“这些条目本该被检索”，只证明在那段窗口内没有人（和没有流程）回头找过它们。

与之对照的是一组下界证据：2026-10-07 之后写入文档目录的材料，由于索引的收集循环从未把该目录的文件放入索引，在当前实现下不可能通过语义检索被取回——这是机制层面的不可达，与模型行为无关。

两个量合起来说明同一件事：在长期运行系统上，“记忆有没有被使用”这个问题的答案，可能根本不是“检索排序不够好”，而是“检索从未发生”或“检索在机制上不可达”。因此评估的第一步不是提高召回率，而是先确认检索通路是否真的接通、是否真的被走过。

由此得到一个记录要求：在统计检索成效之前，必须先把四种情形分开——「没被调用」、「被资源让路而推迟」、「调用了但候选为空」、以及「候选非空但未被采纳」。前两种属于通路问题，第三种属于索引或排序问题，第四种才涉及模型判断。把四者合并成一个「记忆没用上」的比率，会让任何改进都失去着力点。

第 5 阶段的证据上限：检索之后观察到的结果，不等于该条记忆的边际贡献。Rocchi（2026）在 ALFWorld 上做过配对因子审计：若按「检索后成功」记录，169 个被记为贡献的案例里只有 10 个是真正的边际帮助（精确率 5.9%），其余 159 例是把本会成功的案例记到了记忆头上（passenger credit）。这条结果直接限定了本文第 5 阶段能声称什么：只能声称该条被查阅、且计划随之改变，不能声称是它导致的。要把后者做实，需要对同一情境做有 / 无该条目的对照——这属于第 5 章的实验设计，不是观察案例能提供的。

实现层面的对应关系也需要一句交代。本节逐项说明哪些机制来自既有定义、哪些已经实现、哪些尚未确定。与七阶段直接相关的结论是：存储、检索、呈现三个阶段的机制已实现；评估阶段存在判断点但缺少日志；采纳与行为两个阶段尚未实现记录。

3.6 Why These Boundaries Cannot Be Merged　边界为什么不能合并

审稿人最可能问的一个问题是：为什么是七阶段，而不是三阶段、五阶段或六阶段？

本节不试图证明"七"这个数字在某种意义上是唯一正确的划分。它只论证一件较弱、但可检查的事：把相邻阶段合并，会让同一个"记忆没起作用"的诊断指向不同的修法。 因此这些边界不是描述上的细分，而是决策上必要的分界。

三个已确认案例给出直接证据。它们表面现象相同——记忆没有产生影响——但如果只保留"检索 / 利用"这样的粗粒度划分，三者的归属完全相同，而该修的东西在三处完全不同的位置。

案例	现象	粗粒度框架会归为	CMML 判定	实际的修法	误诊的后果
C-07	禁令从未出现在每轮可见的层，TTS 调用属偶发	检索失败或"未使用"	第 3 阶段失效（规则不在注入层）	改注入层配置：把该规则写进常驻层	按"检索失败"处理会去提高召回率，而该规则根本不在检索范围内
C-05	记忆正确且在注入层，两次被逐字引用后仍被判为不可信	利用失败（等同于"被忽略"）	第 4 阶段失效（评估错误）	在判断处留记录，并对条目可信度做复核机制	按"利用失败"处理会去换模型或改提示，而问题在判断环节缺少可回查的证据
C-12	规则被发现、被执行，但条目把两个主体混成一句	利用失败（执行错误）	内容与判断缺陷（位置对、执行对、措辞歧义）	写入时的去重与主体消歧	按"利用失败"处理会去重训或加重提示，而修法在写入侧

三者的共同点是：它们在粗粒度框架下会落进同一个类别，而修正动作分别位于注入层配置、评估环节的记录与条目写入的措辞。三者互不替代——改注入层不会改善 C-05 的判断，改写作措辞也不会让 C-07 的规则出现在眼前。

反方意见与回应。 有人会说：粗粒度框架加一条备注也能表达同样的内容。在个案上这是对的。区别在于能否由规则而不是由注释来判定：备注是写给读者的解释，规则是可以交给另一个人执行的判据。本文选择后者——每个阶段给出判定对象、最低证据与不可判定条件（3.5 节），因此"没有证据"会得到一个明确的结论（不可判定），而不是一句"大概没用"。

这条论证的边界。 本节只用三个案例说明合并会导致误诊，没有穷举所有可能的合并方式，也没有证明不存在更好的划分方式。更强的论证需要把同一批日志交给未参与本研究的判定者，看他们按规则能得出什么结论；本轮未做（见第 6 章与附录 B 的限制说明）。

4. Observable Evidence　可观测证据

本章说明各阶段在这一个被研究系统上能否被观测，以及观测到的失效属于哪一类。本章不主张外部效度：样本是一台机器、一名主要用户、一个长期运行智能体（n=1）。

4.1 Object of Study and Evidence Grading　观察对象与证据分级

> 指代约定：本节与附录 B 在指代被研究的那一个体时用「她」，在指代其运行机制、框架或某一组件时用「系统」或「它」。这一约定只影响行文，不影响证据口径：凡是「她做了什么」的陈述，仍须由系统侧记录支持，不以自述为凭。

Lilith 是本文所研究的长期运行个人智能体：主模型为云端 API（承担理解、规划、判断与语言），运行在本机的 Hermes 运行时之上（该运行时并非作者自建：系统建立在已有的开源运行

时 Hermes 之上，通过补丁、配置与插件做改造——源码补丁集中在 `hermes\patches\`，框架升级会静默覆盖这些补丁，故每次升级后须重打并重核基线 [运行]。本文所称「运行时」一律指这一改造后的运行环境），并由四个本地部署的小模型承担专项工作（本系统的辅助模型目前均为本地部署；部署位置受硬件与合规约束，不构成对辅助模型的定义）——一个视觉模型（读屏幕与图片）、一个 4B 语言模型（批量阅读、摘要、分类）、一个嵌入模型（记忆的向量检索），以及一个更大参数量的本地模型用于长文审读与第二意见。此外存在第二个身份（独立会话表，共享同一套记忆与技能）与两条消息通道。

该系统的实现状态不是本文的主张，而是可审计的观测结果。最近一次系统审计由系统自身完成，其方法与结论均标注了证据等级。审计记录的关键规模如下（均可在本机复核）：

表 5　研究对象的关键规模（2026-10-07 审计值；2026-10-20 复测值见附录 H）

项	取值	证据
会话库	322 个会话 / 217,552 条消息；数据库约 1.3 GB	数据库直读
记忆：常驻注入层	四个文件合计约 1.95 万字符，设有硬性上限，实测贴近上限	在上下文中直接可见
记忆：按需层	十八个文件，顶层合计约 52.6 万字符	目录与文件计数
记忆：归档层	压缩存档、归档、日记、群组、用户档案五类	目录计数
记忆：语义索引	613 个文件 / 12,407 块（构建于审计当日）	索引文件直读
记忆：事实库	202 条	数据库计数
记忆：版本控制	记忆目录为 git 仓库，数百次带说明的提交	git 日志
能力：技能 / 工具 / 插件	技能 193 个；插件约 49 个启用；工具按需加载（53 个候选）	文件计数与配置
调度	框架 cron 22 个（启用 18）；Windows 计划任务 13 个	任务定义与上次运行记录
固定背负	每轮约 93 KB（系统提示约 51 KB + 工具模式表约 42 KB，历史实测）	实测；当前值未重测

审计同时给出十条“本次审计仍无法确认的内容”，包括：压缩由哪个模型承担、若干看门狗是否真的触发过、固定背负的当前值、以及记忆对行为的因果性。本文沿用这种写法：把不确定项显式列出，而不是用推断填补。

4.2 Forms of Failure　失效的形态

把观察到的失效按「规则当时在哪里」归类，比按现象归类更有诊断价值；但仅凭位置不足以覆盖全部观察，因此本节给出两类维度。

4.2.1 按位置分类

未写入（never written）：约定从未进入任何记忆层。2026-09-16 的审批偏好即属此类——用户此前给过「下次你自己点」的口头授权，但系统里没有对应条目，于是在下一次鼠标操作时又触发了一次不必要的询问；该约定在失效后 27 分钟才被写入长期层。

层错位（wrong layer）：规则已经写入，但不在每轮可见的注入层。TTS 禁令写在 `工作纪律.md`，而注入层中 TTS 与语音相关条目为 0；「先看任务清单」只靠索引与脉搏间接提示，没有硬规矩；粗体偏好在 2026-09-16 已进入按需层，两次违反之后（2026-09-26 21:23）才被移入注入层，此后又在两天后因注入层整理而被移出。

在场未执行（present but unexecuted）：规则就在注入层、每轮可见，仍未约束行动。这一类目前没有已确认案例——原先归入此类的浏览器选择案例，经回溯判明是记忆条目措辞含混（把"系统装的浏览器"与"工具驱动的浏览器"混为一句）所致，属内容质量问题，已改列。

分类计数须写清：按位置分类给出三种形态，其中「在场未执行」目前没有已确认案例（理由见 4.2.5）。因此全文若用数字概括，应说「四种已观察形态（未写入、层错位、完成未回写、内容与判断缺陷）＋一种候选但未确认的形态（在场未执行）」，不应笼统称"五类"或"四类"。

4.2.2 按内容与判断分类

位置对了仍然可能失效。观察显示还有两种形态，它们与"放在哪一层"无关：

完成未回写（completion not written back）：动作已经完成，但"已完成"这一状态没有写回任务列表，条目继续以未完成身份存在并被反复重述。C-10 是目前唯一的记录：验收在 10-26 已完成，任务列表 22 天没改，同一句被原样重写 19 次（11 次逐字相同）。它的特征不是"没做"，也不是"忘了做"，而是做了、没人把它标成做完。

内容与判断缺陷（content and judgement defect）：写入了、位置也对，失效出在条目本身或其判断上。已确认三种：同一条规则被写成两份同义条目（冗余）；一条记忆把两个不同主体混在一句里（歧义），后者直接导致了一次错误方案；以及条目正确却被判为不可信（C-05，属判断失效而非内容有误）。三者共同点是：问题不在在哪一层，而在内容被怎么处理。

4.2.3 形态与成因要分开：复述是征兆，不是原因

C-10 的表面形态是「反复重述」：一条待办在 21 天内被原样重写 19 次，其中 11 次逐字相同。但核对日志后，它的成因不是"内容没有变化所以被忽略"，而是验收早已完成、状态没有回写。

这个区分很重要，因为两种解释推出的修法完全相反：

解释	预测	该修什么
重复导致忽略	重复次数越多，被处理概率越低；改变内容应能提高处理概率	增加内容变化、减少重复
状态未回写	条目若被正确关单，复述会自然停止；与重复次数无关	补"完成即回写"的流程

本例支持后者：10-26 验收条件已满足（看门狗三次判定后把网关拉起，`send_ok=True`，用时 25–45 秒，判据 ≤5 分钟），但"已完成"这个状态从未写回条目，于是它继续以未完成身份存在，被每天重述。关单后复述即停止，与"内容是否变化"无关。

因此「反复重述」应记为形态，而不是成因。 它是一个可观测的信号——某条目长期以同一措辞反复出现，说明它的状态可能已经与实际不符，值得去核；但它是线索，不是解释。

由本案例得到的流程缺口是「做完 → 回写」，而不是「减少重复」或「让内容变化」。据此已实施两条改动：关单须写明证据来源；脉搏携带最老条目且同一句三天内不重复。

仍不能确认的：(1) 该条目在这些轮次里是否进入过核心模型的输入——缺少逐轮注入日志；(2) 早间作业重写它时是否读到了"已完成"的迹象——同样缺少记录。这两点使"为什么 22 天没人发现"仍无法完全归因，只能确认"完成状态未回写"这一环节确实缺失。

4.2.4 各形态的修法不同

形态	修什么	观察到的代价
未写入	补存储过程	一次多余询问
层错位	改层的选择策略	规则在两次违反后才被归位
完成未回写	完成即回写：关单须写明证据来源；超期条目提醒	22 天滞留，同一句被重述 19 次
内容与判断缺陷	写入时的去重与主体消歧；对条目可信度的判断留记录	一次错误方案、一轮浪费的测量

把四类合并成「记忆没用」，等于同时放弃四种互不相同的问题。

4.2.5 为什么"在场未执行"这一格是空的

在注入层有硬预算的系统中，常驻内容每轮都在，因此对它而言「检索」与「呈现」几乎同时成立——失效只可能发生在更后面。但这不等于"在场必然被处理"：在场只保证可见性，不保证显著性。显著性是否来自变化，属 F4 的待检验推论。因此这一格之所以空，不是因为它不可能发生，而是因为当前系统无法区分"没看见"与"看见了没动"——两者都需要逐轮注意力记录才能分开，而现有记录装置没有这一层。

四者的修法不同：未写入要补存储过程，层错位要改层的选择策略，完成未回写要补"完成即回写"的流程，内容与判断缺陷要在写入与判断两处留痕——她自己的话是「这是手感不是知识，写进记忆治不了抖腿」。

4.3 Case Set　案例集

附录 B 给出 15 个案例，覆盖七种形态：检索未触发、索引不可达、检索到但未呈现、呈现但未被采纳、拒绝但改变计划、更新引入错误、写入—命中延迟。其中两条是正例而不是失效：一条是索引的机械层缺陷（源码与运行两侧互证，不依赖模型自述），一条是「拒绝但改变计划」——存储的推荐被实施、测量、否决，方案随之改变。

案例集的作用是为阶段判定提供实例，并为后续实验提供任务原型；它不构成效果证据。一条采集纪律需要强调：不要把「模型自述没有采纳」直接当作未采纳的实例，案例必须同时记录当时的原始查询与实际注入内容。

4.4 Data Discipline　数据纪律

本文的所有计数都记录字段口径，因为同一份日志用不同方式统计会得出相差一个数量级的数。

其一，用户角色的消息行中约 22%（6,065 / 27,293）并非用户发言，而是上下文压缩交接块、网关来源标注与 cron 通知；按关键词统计用户纠正次数会显著虚高。其二，按文本匹配工具名字段会命中工具定义与脚本正文——2026-09-28 一天可虚增至 503 行，而按 `tool_name` 精确统计，该检索工具在 2026-09-30 至 10-05 之间只有 16 条记录、10 个不同调用时刻。

其共同点是：在日志型数据上，字段语义比字符串匹配可靠，一条记录未必是它所声称的那件事。这也是本文主张「阶段判定以注入日志、检索记录与工具参数等系统侧证据为准，不以模型自述为准」的另一个理由。各项数值的测量日期与复测结果见附录 H；操作定义、采集语句与逐项可复现性判断见附录 I。

4.4.1 五名外部判定者的一致性与规则缺口

这一步要检验的是规则能否被他人执行。阶段判定若只由作者完成，规则是否可被复现就无从判断。本文把判定规则与案例证据字段封装成一份材料包，交由若干未接触原始日志、也未见过本文记录判定的外部判定者独立判定，再计算一致性。

材料包包含 3.5 节的规则表（含 R1 至 R10 十条裁定，其中 R1「最小修复点」优先于其余）与 15 个案例的时间、经过、证据三个字段。案例顺序重排、标题删除，与正文 C-01 至 C-15 的对应关系只有作者掌握。判定者输出「最早失效阶段」，并对每条给出「已确认 / 暂定 / 不可判定」的结论强度。

判定者共五名，全部为人工智能系统，也全部满足盲判条件：GPT-6 Luna、Qwen3.8 Max、DeepSeek、文心 5.1、GLM-5.3。各判定者的版本与推理配置并不统一，其中部分未公开具体配置；因此一致性只能解释为规则可被独立执行的程度，不等同于人类标注者之间的信度。本文作

者与本文使用的助手都不参与 κ：前者的记录判定形成于规则成文之前，后者参与过材料整理并读过已有结论。

结果是两两 κ 为 0.688 至 1.000，Fleiss' κ = 0.840；15 个案例中 11 个五人完全一致，按多数票与记录判定比较则是 14 个一致。

四处分歧之中三处是单一判定者离群：DeepSeek 把 C-08 判为第 4 阶段失效；GPT-6 GPT-6 Luna 把 C-09 判为第 1 阶段，并对 C-01 拒判为不可判定。第四处是真正的规则平票——C-15 上两名判定者判第 2 阶段、两名判第 3 阶段、一名判不可判定。它暴露 R2 与 R9 的边界尚未闭合：条目「能查能读」但没有硬规矩触发去查，按 R9 属检索未触发，按 R2 又因它不在每轮注入层而属呈现缺失。本文不在此处强行裁决，把它列入 6.5 节的第一项。

规则也约束了作者自己。C-02 是唯一一例多数判定推翻记录判定的案例：五名判定者一致判「不可判定」，理由是证据只有文件时间戳，按 R10 不足以支撑「未触发检索」这一否定性结论。本文据此修改了该案例的判定（见附录 B），并列出所需的补充记录。判定规则在成形之后推翻过作者自己的结论，不是事后包装。

局限有三。其一，判定者全部是人工智能系统，κ 的含义限于规则的可执行性，不能替代人类双人标注。其二，五名判定者的版本与推理配置并不统一，且部分未公开，一致性可能部分来自共同的规则约束、部分来自不同的算力，本文不区分这两者。其三，材料包中的证据字段是本文整理的二手材料，判定者看不到原始日志，因此一部分分歧来自材料本身的信息损失，而非规则本身。

4.5 Measurement Method　测量方法

本节说明本文所用的测量口径。完整的操作定义、采集语句与逐项可复现性见附录 I。

四条通则。

其一，计数一律按字段，不按文本。工具调用以 `tool_name` 字段精确匹配统计；用文本模式匹配会命中工具定义与脚本正文，同一天的计数可相差一个数量级。

其二，用户消息须先剔除系统块。被研究系统的 `role='user'` 消息中约有 22% 并非用户发言，而是上下文压缩交接块、网关来源标注与定时作业通知。不剔除这些块，任何"用户纠正次数"都会显著虚高。

其三，时间一律换算到本地时区再比较。数据库时间戳为 UTC，直接以本地时间字符串作条件会错开八小时；这一坑在本轮核查中实际发生过一次，导致一个时间窗口被误判为"无记录"。

其四，阶段 4 与 5 的判定须与下游产物互证。推理轨迹（reasoning_content，覆盖 97.8% 的助手消息）只能证明模型"说了什么"；只有当它与注入记录、工具参数或文件差异一致时，才升级为已确认；两者矛盾时以下游为准。

两项指标的定义。

· 复述率 = 原样重写次数 ÷ 该条目出现次数。用于度量"写了但没变"。

· 提问次数 = 该条目引发的、由模型主动向用户发起的询问次数。这是一个二值、可数、可审计的事件，用于度量提示是否产生了交互。

一项已撤回的方法。 早期曾用"取一个该问题仍未解决的时间点，减去写入时刻"来计算写入—命中延迟。该方法有两个错误：把"未解决"当作"未被使用"，以及把相邻事件误当同一件事。正确做法是读取条目自身的出现史（创建、每次重写、每次状态变更），只有出现"长期零出现且无外部阻塞记录"的区间才构成延迟。按此重测，原记的约八天延迟不成立，已撤回。

5. Evaluation Design　评估设计

本章给出可执行的检验路线。它在方法提案与已完成工作之间的位置需要说明：这是一份设计，不是结果。

5.1 Frozen Core, Evolving Workspace　冻结核心与演化工作空间

当前可操作的路线是 Frozen Core + Evolving Workspace（冻结核心模型 + 演化工作空间）。在该设置中，核心模型参数保持不变，工作空间中的记忆、知识、经验、技能和组织方式可以根据任务反馈逐步更新。该路线适用于无法直接训练或修改所用 API 模型参数的条件。

「演化」必须被操作化为可计数的事件，否则无法与「文件变多」区分。本文建议以以下事件类型作为演化计量单位：新增事实记忆、修正事实记忆、删除过时记忆、合并冲突条目、更新经验摘要、调整技能说明、改变资源组织方式。仅文件数量增加，不等于工作空间有效演化；更新也可能引入冗余、错误或过时信息。实验报告应同时给出各类演化事件的数量与净效应（新增与修正的比例）。

未来若具备可控的模型参数更新条件，可进一步研究 Core–Workspace Co-Evolution（核心模型—工作空间共同演化）。概念上可表示为：

$$C(t+1) = F(\, C(t), W(t), Fb(t)\,)$$

$$W(t+1) = G(\, W(t), C(t), Fb(t)\,)$$

其中 C(t) 表示时刻 t 的核心模型状态，W(t) 表示工作空间状态，Fb(t) 表示任务结果或外部反馈。该表达式只是概念性记号，不是已经定义或估计的数学模型，也不意味着当前系统能更新模型参数。

后续可以比较以下四种条件：

5.2 Metrics and the Minimum Viable Experiment　指标与最小可行实验

工作空间的演化可以按资源类型分别描述：记忆从候选形成、整理与审核、保存、检索与使用、效果反馈到修订或归档；经验从原始经历经结果与原因分析提炼为可复用方法、约束或建议；经验与知识进一步参与技能与工具的形成与改进。并非所有经历都应转化为技能，成功案例也不必然意味着其方法在其他情境下同样有效。各类更新的具体规则、触发条件与效果仍待定义与验证。

为降低首轮实验的成本，「最小可行实验」不应同时检验全部五个研究问题。本文建议首轮实验只检验一个命题：在冻结核心模型条件下，事实记忆与会话记录联合使用，相对仅使用事实记忆，是否提高了记忆的决策影响率（第 5 阶段）与行为有效率（第 6 阶段）。同时把“写入—命中延迟”（3.5 节）作为主要指标之一，因为它可以在不构造完整对照任务集的条件下先行采集。下表给出该命题所需的实验要素。

表 6　最小可行实验设计（方法提案）

要素	设定	理由
核心模型	固定版本与参数，记录版本号与温度设置	冻结核心是路线前提；版本变化会同时改变所有条件
任务集	30–50 个可重复任务，覆盖至少两类任务族	支持配对比较与两类任务族间的外部效度检查
条件	M0 无记忆（对照）/ M1 仅事实记忆 / M2 事实记忆 + 会话回溯 / M3 全量演化工作空间	对应本文的互补关系主张与附录 D 的 A/B 条件
重复	每个任务 × 条件重复 3 次，随机化执行顺序	估计随机性方差；顺序随机化避免环境漂移混淆
主要指标	决策影响率（第 5 阶段成立的任务比例）、行为有效率（第 6 阶段成立的比例）	直接对应 1.2 节的命题与 3.5 节的阶段判定；避免以任务成功率为唯一终点。写入—命中延迟不列为主要指标：其操作定义尚不足以支持统一口径（"命中"指检索、注入、提及还是下游使用？多次命中如何计？有外部阻塞的条目如何删失？），须先冻结事件定义与删失规则
次要指标	检索命中率、注入 token 占用与固定背负、上下文长度、延迟、工具调用次数、线索粒度档位、冗余与冲突条目比例	用于解释主要指标变化来源与成本

要素	设定	理由
分析	配对检验（同一任务跨条件比较）；报告效应量与 95% 置信区间；预注册主要指标	配对设计提高功效；预注册避免事后挑选指标
探索性指标	复述率、提问次数、写入—命中延迟	前两者口径明确（附录 I.2）；延迟须先冻结定义，暂作过程观察
噪声控制	按表 D-2 分类记录工程 / 能力 / 调度失败，并单独报告三者比例；正确拒绝不计入失败	避免把运行时噪声读成记忆机制的效应
记录要求	每次检索、注入、取舍与更新的原始日志	使第三方能够独立重跑第 3.5 节的判定规则

关于统计功效：若主要指标为二值判定（采纳/未采纳），在配对设计下需要足够数量的任务—条件配对才能检出中等效应；30–50 个任务仅为下限，正式实验应根据预实验估计的效应量重新计算样本量。若只报告点估计而不报告区间，结论无法与噪声区分。

本系统已具备部分指标的基础设施，可直接复用：记忆目录本身是 git 仓库，因此写入时间戳与演化事件（新增 / 修正 / 删除）可逐条取回；常驻注入层有明确的字符计数与上限，因此“呈现”阶段的成本可直接读出。缺少的是采纳层与行为层的记录，这也是后续工作的第一优先项。

· 内部效度：检索策略与记忆内容同时变化时，无法区分效应来源；应分别消融检索与内容。

· 构念效度：以任务的最终成功率代表「记忆有效」，会把工具失败与推理失败计入记忆效果；应以阶段指标为主。

· 外部效度：单一实践对象、单一模型版本与单一任务族的结果不能直接推广（见第 6 章）。

· 结论效度：多次比较同一任务集会增加 I 类错误风险；应预注册指标并作相应校正。

5.3 Recording and Reproducibility　记录与复现

要复现本文所描述的方法与观测，至少需要：核心模型的版本与解码参数；运行时的版本与本地改动；记忆层各文件的版本；检索调用的原始查询串与返回结果；注入内容的实际构成与预算占用。当前可提供的是后三项中的部分记录；前两项在本文写作时未完整记录，属于已知限制。

另需注意两类计数陷阱，它们会让「同一份日志」得出不同的数：其一，用户角色的消息行中约 22%并非用户发言，而是上下文压缩交接块与系统通知，按关键词统计用户纠正次数会显著虚高；其二，按文本匹配工具名字段会命中工具定义与脚本正文，与真实调用相差一个数量级。凡涉及计数的结论，都应在报告中标明字段口径与去重规则（见附录 H）。

6. Discussion　讨论

本章讨论这套分解的解释力、适用范围、局限与后续工作。

6.1 Interpretation　解释

存储容量、检索性能和最终任务表现不是同一指标。系统可能检索准确却未把结果呈现给模型；可能呈现了记忆但模型判断其不适用；也可能使用了记忆却因工具失败而未能完成任务。对这些阶段分别记录，有助于定位失败发生在哪里，也决定了改进应发生在哪一层：检索层的问题不能通过扩大存储解决，采纳层的问题不能通过提高召回率解决。

6.2 Relation to Existing Work　与既有工作的关系

历史记录并不天然可靠。总结可能遗漏条件，成功经验可能依赖特定环境，失败记录也可能包含错误归因。因此工作空间演化需要考虑来源、时间、适用范围、不确定性、冲突和修正机制。更多记忆不一定改善行为，甚至可能增加误导风险：当上下文中同时存在相互冲突的条目时，模型需要额外能力来发现并解决冲突，而这一能力本身不应被假定存在。

6.3 Applicability and Boundaries　适用范围与边界

如果所有控制都依赖模型主动遵守提示，运行时将难以在重复调用或无进展时提供可靠约束；但运行时过度介入也可能打断合理探索。因此监督机制应以可观测条件、明确权限和分层干预为基础，并通过实验评估误中止率、资源消耗、任务成功率及恢复能力。其中误中止率应作为一等指标：一个频繁打断正确探索的监督器，其代价可能高于它避免的资源浪费。被研究系统在资源层提供了一个具体的平衡样例：当模型请求使用本地模型而显卡正被用户占用时，运行时会直接拒绝并给出原因。这种拒绝在直觉上像一次失败，实际上是一次成功的运行时行为；把它计入失败率会系统性低估系统，把它忽略不计又会掩盖真实的资源冲突。因此“拒绝”必须作为独立事件类型记录，而不是二值地归入成功或失败。

6.4 Limitations　局限

先说材料性质：这不是短板，是另一种取舍。既有工作使用受控基准（LoCoMo、ALFWorld、生成的客服流、元认知题库），因而可比、可复现、能做反事实归因；但这份"干净"是把环境简化到可控程度换来的——单用户、无并发、无预算压力、无长时间无人值守的状态漂移。本文的材料相反：一个真实个人系统的日常连续记录（401 个会话、227,477 条消息、技能 193–200 个、定时作业 22–23 个、索引 613 个文件 / 12,407 块），失败带真实代价（C-05 中 11

个文件被错误改写；C-13 中一条授权偏好缺失 27 分钟；C-10 中一条已完成的待办滞留 22 天）。

关键在于：本文观察到的形态需要时间、并发与人的中断共同作用才会出现——层错位要求同一条规则分布在不同可见层而判断发生在另一层；完成未回写要求动作已完成、状态却没有回写，且需多天重述才暴露；复述而零推进要求同一句在每日作业里被反复写入。这些现象不是"数据量不够"所以基准里没有，而是基准没有那个时间尺度与使用密度：单次实验里，状态漂移来不及发生。

代价同样明确：本文不能做反事实归因。第 5 阶段只声称"该条被查阅、且计划随之改变"，不声称"是它导致的"；补上这一层需要在同一系统上做有 / 无对照，属于第 5 章的实验，不属于本文已完成的观察。

长期运行还带来一个混杂，需要单独讨论：用户越来越了解系统，系统也越来越了解用户，因此"长期使用后表现变好"可能来自模型能力、记忆积累、运行时改进、用户提问方式的改进、用户行为本身的改变，以及双方的相互适应。由此提出的纪律是——「长期使用后性能提高」不自动等于「模型自身能力提高」；在冻结核心模型的路线下，任何观测到的改进都必须先排除用户侧的变化。研究对象里的具体形态是：被观测到的行为变化（新增的资源层、维护层与岗位编制）来自设计变更，不是系统自行生长；把设计变更与自发演化分开记录，是本节的最低要求。

研究对象所基于的 Hermes Agent 是一个开源智能体运行时，本文在其上做了本地源码修改，但本文不提出新的运行时架构：运行时的角色被限定为事实与权限的提供者，它本身的机制设计不在本文的主张范围内。本文讨论的两项机制（认知层的分层监督、机器人环境接口）目前确实没有完整可运行的实现，阅读时应计入。

表 7　CMML 各阶段的隐私与安全风险及对应措施

阶段	主要风险	对应措施（待实现与评估）
1 Stored	过度采集；保存第三方信息；无保留期限	数据最小化；字段级脱敏；保留期限与自动过期
2 Retrieved	跨上下文泄露；检索放大敏感内容	按用途限定检索范围；敏感条目分级与访问控制
3 Presented	敏感信息进入不可控上下文（含外部 API）	注入前过滤；对出站模型调用做敏感字段屏蔽
4 Evaluated	被投毒条目被判定为适用	来源可信度标记；写入需可追溯；对异常来源条目降权
5 Decision-influential	采纳错误记忆导致错误行动	高风险动作前的确认点；采纳记录可审计
6 Behaviorally effective	不可逆外部副作用被放行	副作用工具的权限模型与幂等/可逆性标记（见 3.2 节）
7 Updated	投毒持久化；错误归因被写入；无法删除	写入审计；修正与删除机制；冲突条目的显式标记

记忆投毒需要单独强调：记忆写入是一条可在会话中触发的持久化通道，一条被写入的虚假事实会在后续所有相关任务中被检索、评估并可能被采纳，其影响时间长于单次提示注入。「评估」阶段同样薄弱：若模型倾向于采纳带来源标记的历史条目，则该标记本身成为攻击面。本文把「记忆是否可被追溯、可被修正、可被删除」视为实现前提，而不是可选功能。

本文的边界与待办如下。前三条属于自我设定的范围，不是待补齐的缺陷；后一条属于真正的局限。

· 文献比对的范围有限：20 条参考文献的出版信息已逐条比对 DOI 与 arXiv 注册记录，表 E-1 涉及的各篇另核到摘要页或正文相应章节；但系统检索尚未进行（第 2 章只覆盖笔者判断最接近的一组工作），各文献的实验设置与完整方法未逐篇通读，中文文献未检索。

· 范围：不报告受控实验。3.5 节的判定规则经五名外部判定者独立检验（4.4.1 节，Fleiss' κ = 0.840，11 例完全一致），但判定者全部是人工智能系统、推理强度不齐，C-15 上仍有 2:2 的规则平票；人类双人标注的 κ 仍待进行（4.4.2 节）。

· 实现状态：逐项矩阵见 3.5 节。仍缺的是案例规模——已按附录 B 的格式整理 15 个案例（含 1 个未核实案例），尚不足以支撑统计。

· 通用性尚待检验（真正的局限）：单一实践对象、单一模型版本的设计经验不能直接推广到其他模型、框架或机器人系统。此外，观测对象只有一个，且研究者同时是系统开发者与使用者，存在观察者偏差；阶段判定的主语一律取系统侧；使用者同样持有、判断并改写这些记忆（她的纠正与授权会改变条目的最终形态，附录 B 中有若干案例），但使用者的判断未纳入本文的框架。缓解：在至少两个任务族与两个模型版本上重复实验，并对关键结论保留原始日志以便他人复核。

6.5 Future Work　未来工作

后续工作应按依赖关系排序，而不是并行扩展所有方向。

1. 阶段定义与标注验证。 先冻结七阶段的操作性定义，尤其明确「评估」「决策影响」与「行为有效」的证据边界；并闭合 R2 与 R9 尚未解决的边界——条目位于不进每轮注入的层时，若「能被看到」依赖外部动作（打开文件、发起检索）判第 3 阶段，若条目本身在每轮可读范围内、只是缺少触发动词判第 2 阶段。规则补入后需在新一批案例上重测。以一组去标识化运行案例进行双人独立标注，报告一致性、分歧类型及修订过程；本文目前只有一名人类研究者，第二名标注者尚不具备，因此现阶段先公开材料包与判定结果，供第三方按同一程序复现。

2. 补齐事件级记录。 为检索调用、候选集、实际注入、模型对候选的处理、下游计划变化、工具结果与记忆更新建立可关联的事件标识。日志至少记录时间、任务 ID、候选记忆 ID、阶段事件、结果与失败类型；敏感内容应最小化记录并按策略脱敏。

3. 完成最小对照实验。 固定核心模型版本、提示模板、工具与任务集，比较无记忆、仅事实记忆、事实记忆加会话回溯等条件。先用预实验估计效应与方差，再确定正式样本量；写入—命中延迟作为过程指标，不能单独作为记忆有效性的证据。

4. 核验研究对象与文献。 将关键运行事实关联到可复核的源码提交、日志或快照；对相关工作逐篇核实原文、出版状态和比较表中的判断，并记录检索日期与纳入标准。

5. 最后再扩展架构主张。 在 CMML 的判定规则和实验路径稳定后，再讨论机器人场景与核心模型—工作空间共同演化问题，避免把尚未实现的方向误读为本文已经验证的能力。

6. 按证据等级标注案例。 现有案例的证据完整度差异很大，从注入快照到仅剩时间戳不等。拟为每个案例标注证据等级（E1 全记录 / E2 中记录 / E3 弱记录 / E4 仅自述），并据此限定可判范围——没有检索调用日志时不得判「未触发检索」，只能判「未使用·阶段不可判定」。

这一路线把近期工作分为“先让概念可被他人稳定判定”与“再检验机制是否带来收益”两步。若第一步失败，应先修改阶段划分，而不是直接扩大实验规模。

7. Conclusion　结论

本文的核心主张是：智能体记忆的关键问题不是存储容量，而是记忆能否在适当时机进入当前认知过程并改变行为；因此记忆评估必须区分保存、检索、呈现、评估、采纳、行为有效与更新七个阶段。为把该主张变成可检验的对象，本文提出 CMML，并为每个阶段给出可观测事件、判定规则与典型失效模式；并在一个基于 Hermes Agent（开源运行时，含本地源码修改）的个人智能体上，把每个阶段落到可观测的记录上。

在此框架下，本文主张候选记忆的适用性应由核心模型判断，事实记忆与会话记录构成互补双层，成功与失败经验都应保留但须附带适用条件，运行时的执行控制应与认知判断分层并对具有外部副作用的动作单独定义语义。对于当前无法直接更新核心模型参数的研究条件，冻结核心模型、演化工作空间提供了一条可操作的实验路线，本文给出了其对照组设计、最小可行实验与需要控制的变量。

本文属于预发行阶段的概念与方法稿：它提供一套描述记忆如何进入认知过程的概念、一组给出操作性判定规则的阶段定义、以及一份可执行的实验设计，但不提供效果证据。按本文自身提出的标准，这份分解在下列条件下应当被修正：（1）若“写入—命中延迟”在受控条件下不可复现，则 1.2 节的机制命题不成立；（2）若七个阶段中的若干阶段无法被独立标注者稳定判定，则该分解过细，应合并；（3）若检索、呈现、采纳三阶段的衰减在冻结核心的对照实验中不可测出，则“阶段化评估”相对粗粒度层级区分没有增量价值。

这三条都是可操作的：它们各自对应一个具体的测量，而不是立场之争。本文希望被这样使用——把它当作一份可以被证伪的问题清单，而不是一份结论。

References　参考文献

[1] Yao S, Zhao J, Yu D, et al. ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023. arXiv:2210.03629

[2] Shinn N, Cassano F, Berman E, et al. Reflexion: Language Agents with Verbal Reinforcement Learning. NeurIPS 2023. arXiv:2303.11366

[3] Wang G, Xie Y, Jiang Y, et al. Voyager: An Open-Ended Embodied Agent with Large Language Models. arXiv:2305.16291

[4] Park J S, O'Brien J C, Cai C J, et al. Generative Agents: Interactive Simulacra of Human Behavior. UIST 2023. doi:10.1145/3586183.3606763

[5] Mei K, Zhu X, Xu W, et al. AIOS: LLM Agent Operating System. COLM 2025. arXiv:2403.16971

[6] Xia Y, Fan J, Chen W, et al. AgentRM: Enhancing Agent Generalization with Reward Modeling. ACL 2025. doi:10.18653/v1/2025.acl-long.945

[7] Zhang Y. Agent libOS: A Runtime Substrate for Capability-Controlled Self-Evolving LLM Agents. arXiv:2606.03895

[8] Sumers T R, Yao S, Narasimhan K, et al. Cognitive Architectures for Language Agents. TMLR 2024. arXiv:2309.02427

[9] Xu W, Wang Y, Mei K, et al. MemGym: a Long-Horizon Memory Environment for LLM Agents. arXiv:2605.20833

[10] Yu M, Hu D. VitaBench: Memory Lifecycles Beyond Retrieval. arXiv:2610.14731

[11] Lin Z, Hao X, Fu R, et al. A Survey on Long-Term Memory Security in LLM Agents: Attacks, Defenses, and Governance Across the Memory Lifecycle. EMNLP 2026, 52221–52235. aclanthology.org/2026.emnlp-main.2296

[12] Song Y, Wang Y, Ma X, et al. Retrieval-Driven Memory Reconsolidation for Long-Term LLM Agents. arXiv:2609.16053

[13] Yuan B, Su Y, Yao K. Diagnosing Retrieval vs. Utilization Bottlenecks in LLM Agent Memory. arXiv:2603.02473

[14] Rocchi A. Does Memory Credit Travel? Paired Factorial Audits of LLM-Agent Memory. SSRN 2026（预印本，未经同行评审）. doi:10.2139/ssrn.7160321

[15] Nelson T O, Narens L. Metamemory: a theoretical framework and new findings. The Psychology of Learning and Motivation, 26: 125–173, 1990. doi:10.1016/S0079-7421(08)60053-5

[16] Cacioli J-P. The Metacognitive Monitoring Battery: A Cross-Domain Benchmark for LLM Self-Monitoring. arXiv:2604.15702

[17] Nader K, Schafe G E, LeDoux J E. Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. Nature, 406: 722–726, 2000. doi:10.1038/35021052

[18] Zhang Z, Dai Q, Bo X, et al. A Survey on the Memory Mechanism of Large Language Model-based Agents. ACM TOIS, 43(6): 155, 2025. doi:10.1145/3748302

[19] Packer C, Wooders S, Lin K, et al. MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560

[20] Xu W, Liang Z, Mei K, et al. A-MEM: Agentic Memory for LLM Agents. NeurIPS 2025. arXiv:2502.12110

以上 20 条于 2026-10-28 逐条比对 DOI 与 arXiv 注册记录。比对中修正三处：Reflexion 的作者表补入 Berman；MemGPT 的年份由 2024 改为 2023；AgentRM 的会议届次由 61st 改为 63rd（ACL 2025 为第 63 届）。为控制篇幅，每条只列前三作者与一个主要标识符，完整作者表与出版信息见各文献原文。

Appendices　附录

Appendix A. Terminology Table　术语表

表 A-1　术语对照表

中文	英文	本文定义要点
Hermes Agent	Hermes Agent	研究对象所基于的开源智能体运行时，本文在其上做了本地源码修改
CMML	Cognitively Mediated Memory Lifecycle	记忆生命周期，七阶段及其判定规则，见 3.5 节
核心模型	Core Model	承担主要认知任务并判断候选记忆适用性的模型角色
辅助模型	Auxiliary Model	承担专项能力（检索线索、表示、图像生成与处理等）但不承担最终采纳判断的模型角色
事实记忆	Memory Facts	紧凑、可检索的事实或摘要，须承载适用条件
会话记录	Sessions	原始交互记录，保留语境与来源，基本不改写
决策影响	Decision-influential	记忆内容影响了下游可观察产物，见 3.5 节判定规则
行为有效	Behaviorally effective	在对照条件下，采纳与否导致行动或结果出现差异；反事实判断
工作空间演化	Workspace Evolution	以七类可计数事件为计量单位的更新过程，见附录 D

本附录给出观察案例集的记录模板。该模板需在正式稿公开前完成填写；本修订版不包含案例数据，也不以示例数据代替。

Appendix B. Observation Case Set　观察案例集

本节给出 15 个被记录的观察案例，覆盖第 3 章七阶段中可能出现的失效形态与两个正例。样本与外效的范围同第 4 章。案例编号与正文第 3.5 节的判定规则一一对应。

B.1 脱敏处理　De-identification

公布案例之前，观察材料经过一次脱敏。处理只有三项，都以保证事实可核对为前提。

其一，会话标识保留时间戳、去掉末尾的随机标识符，按首次出现顺序替换为字母（a、b、c……）。读者仍能判断事件发生在哪一天哪一分钟、同一次会话是否被引用两次，但无法据此在自己的系统里反查。原始标识与字母的对应表由作者保留。

其二，文件系统路径改为工作区内的相对路径，图件改为随文附带，不保留作者机器上的目录结构。

其三，不收录第三方个人信息、账号标识、通信录与任何凭据。案例中出现的对话片段限于判定所需的最小范围，均已确认不含可识别到具体个人的信息。

脱敏没有改动任何供判定使用的事实：时间、阶段、证据类型、计数与结论强度全部按脱敏前的口径保留。这一点可以对照 4.4.1 节自行检验——外部判定者拿到的就是脱敏后的材料，他们的判定与作者的记录判定在 14 个案例上一致。

B.2 数据与材料可得性　Data and Materials Availability

复现本文 4.4.1 节一致性检验所需的材料已整理成一份独立的材料包，可向作者索取。内容包括：判定规则表与 15 个案例的证据字段（即判定者看到的盲判材料包）、五名判定者各自的原始输出、材料包编号与正文 C-01 至 C-15 的对应表、以及判定者登记（模型版本、推理配置与盲判条件）。统计口径（阶段编码 1–7、非失效 0、不可判定 −1）与两两 κ、Fleiss' κ 的计算方式一并附在该包内。

获取方式：向作者索取（通讯邮箱见首页）。作者会提供一份压缩包，内含判定材料包、五份判定输出、编号对应表与判定者登记，并附统计口径与 κ 的计算方式。

未公开的部分是原始运行日志、会话数据库、真实会话标识符与第三方信息。材料包经过脱敏（见附录 B.1）：会话标识只保留时间戳，末尾随机串按出现顺序替换为字母；文件路径改为相对形式；不含账号与凭据。脱敏不改动任何供判定使用的事实。

本文观察对象的完整运行环境（Hermes Agent 运行时、本地补丁与插件配置）不在公开范围内：它是作者的日常运行系统，包含私人数据。读者可据 3.5 节的判定规则与附录 B 的案例字段独立复核判定结论；如需重跑第 5 章设计的实验，应以自己搭建的环境为准。

B.3 证据分级约定　Evidence Grades

[源码] 源文件与行号可直接核对 · [运行] 日志 / 记录 / 快照 / 数据库 · [实测] 可重复测量 · [自述] 仅来自系统自身陈述 · [推断] 间接推定。仅含 [自述] 的条目不因补充上下文而升级。

B.4 覆盖情况　Coverage

表 B-1　观察案例的覆盖情况（含改判后状态）

案例	现象	位置轴	内容轴	判定状态
C-01	技能在库中从未被加载	未写入通路	—	部分确认（加载通路未触发）
C-02	群档案有答案却逆向读源码	—	—	未核实（不采信）
C-03	索引收集循环缺 append	层错位（机械层）	—	已确认
C-04	索引陈旧七天、重建后零调用	未走过	—	已确认

案例	现象	位置轴	内容轴	判定状态
C-05	判错一条已在注入层的正确记忆	—	判断失效（内容本身正确）	已确认（推理 + 下游 diff）
C-06	注入层裁剪移出 19 条	层错位	—	已确认
C-07	禁令从不在注入层	层错位	—	已确认
C-08	自己两篇笔记结论相反	—	内容缺陷（冲突）	已确认（推理 + 原始输出）
C-09	同一条规则写了两遍	—	内容缺陷（冗余）	已确认
C-10	早已验收完成却未关单，19 次原样重述	—	完成未回写（形态为维持性复述）	已确认（git + 看门狗日志）
C-11	粗体偏好两次违反后才归位	层错位	—	部分确认
C-12	记忆条目混了两个主体	—	内容缺陷（歧义）	已确认（推理）
C-13	审批偏好从未写入	未写入	—	部分确认
C-14	核心记忆 2200 字的用户侧证据	—	—	已确认（作为容量锚点）
C-15	任务清单存在但未查	层错位	—	部分确认

按四类失效汇总：

形态	案例	数量
未写入	C-13（部分）	0 已确认
层错位	C-03、C-06、C-07（+C-11、C-15 部分）	3 已确认
完成未回写	C-10	1 已确认
内容与判断缺陷	C-05（判断失效，内容正确）、C-08、C-09、C-12	4 已确认
维持性复述（形态）	C-10 的表现形式；成因为完成未回写	1 已确认
在场未执行	—	0（见 4.2.5）

C-02 保留在册但标注"未核实、不采信"：它既无会话 id，也无源码或日志佐证，仅有一条二手描述。保留而非删除，是为了让读者看到案例筛选的标准被执行过。

C-01 检索未触发：技能库里有工具，而它甚至预言了这次错误

· 时间：事件 2026-09-26；记录首次提交 `12c020f`，2026-09-27 00:20:14

· 经过：智能体需要定位一批解包后的资源文件，自行写了一个 Python `os.walk` 递归扫描，耗时 96 秒。用户问「es.exe 你不是搞过吗」后，`es.exe` 用 0.306 秒给出同样结果（约 314 倍差距）。系统技能库中已有 `windows-file-inventory` 技能，首行即写着「别扫盘」，但该技能从未被加载——因为加载判定依据是「我正在干什么」，而「找一个文件」没有被当作一件任务

相关记录：技能文件 `skills/devops/windows-file-inventory/SKILL.md` 第 16 行；`经历.md` 第 167–174 行的根因说明。

· CMML 阶段：检索失效（阶段判定成立，强度暂定——「从未被加载」这一否定性事实主要由自述支撑，见 4.4.2 节）

· 证据：[源码] 技能文件行号；[运行] 索引文件中的该技能条目；[自述] `经历.md` 根因分析

· 脱敏：无个人或第三方信息

C-02 检索未触发：答案已存在群组档案里，智能体却去逆向读源码

· 时间：2026-09 下旬（具体时刻见原始记录）

· 经过：同一个问题在过去某次群聊中已经得到过答复，且该答复已写入按群归档的记录文件；智能体没有检索该记录，而是逆向阅读框架的适配器源码重新推导结论

· CMML 阶段：不可判定（五名外部判定者一致认为：仅有文件时间戳，按 R10 不足以支撑「未触发检索」，见 4.4.1 节）

· 证据：[运行] 群组档案文件与源码文件的时间戳

· 脱敏：本案例涉及第三方数据。群标识一律写作 `groups/<GROUP_ID>/`，群成员写作 `[群成员A]`；不引用原始查询串

C-03 索引不可达：收集循环从不把文件加入索引

· 时间：索引构建于 2026-10-07 13:37:30；缺陷位置在 `scripts/memory-search.py`

· 经过：文档目录的收集循环逐条过滤候选文件（跳过临时目录、超出体积上限等），但没有任何语句把通过过滤的文件加入待索引列表；紧随其后的技能目录循环则正确地执行了加入操作。后果是文档目录下的文件从未进入索引。以索引文件实测：613 个被索引文件中，文档目录 0 条，群组目录 0 条

连带后果：2026-10-02 注入层裁剪时移出的 19 条记忆中有 3 条被「迁移」到文档目录下的一个备忘文件（16,614 字节）——而该文件同样不在索引中，迁移的逃生路线并不存在。

· CMML 阶段：Retrieved 在机械层失效。Stored ✓（文件在磁盘上），但对检索工具不可达

· 证据：[源码] `memory-search.py:139-147`（无加入语句）与 `:148-160`（有加入语句）

· [运行] 索引文件的 613 条目录统计

· 备注：此前一份观测记录曾据源码注释判定「文档目录自 2026-10-07 起已被覆盖」，该判断只读了常量定义、未读消费循环，与本案例结论相反。本文采用源码与运行两侧一致的结论

· 脱敏：只涉及本地路径与磁盘容量，无个人标识

C-04 索引不可达：一份陈旧索引服务了七天，重建后的索引却从未被查询

· 时间：索引 2026-09-28 02:01:48 构建（337 块）→ 2026-10-07 13:37:30 重建（12,407 块）

· 检索最后一次调用 2026-10-05 04:34:03

· 经过：从 09-30 到 10-05 的每一次语义检索，返回结果的表头都重复同一行「索引建于 2026-09-28 02:01:48，共 337 块」——意味着至少七天里所有语义查询都由一份冻结快照作答，此后写入的内容对检索不可见。索引最终在 10-07 重建，但检索工具在 10-05 之后一次也没有再被调用：新索引从未被使用过

· CMML 阶段：Retrieved 双重失效——先是陈旧（七天对着冻结快照检索），后是闲置（重建后十三天零调用）

· 证据：[运行] 检索记录的结果表头（四条独立记录互证）

· 索引文件的 `built` 字段与文件时间

· 数据库按工具名精确统计的调用分布（09-30 共 8 次、10-01 共 2 次、10-04 共 2 次、10-05 共 4 次，末次 10-05 04:34:03）

方法论备注：按工具名精确统计得 16 行；若用文本模糊匹配统计同名字段会得到数百行——其中大部分是工具定义文本与脚本正文，不是调用。本文所有调用计数一律按工具名精确统计。

· 脱敏：检索记录中包含第三方姓名与个人对话，此处只引用时间戳与表头行，不引用查询文本

C-05 判断失效：记忆正确且在注入层，却被判为不可信

· 时间：事件 2026-09-26 22:37

· 记录提交 `12c020f`，2026-09-27 00:20:14。会话 `20260923_024146_a`

· 经过：系统在一个名称「莉莉安」上判断错误——它认定这个名字是自己编造的，并在 11 个文件里把它改成了「千问」。该名称实为使用者给本地模型起的名字，正确条目就写在注入层（「本地模型：副手莉莉安」）。推理轨迹显示：22:36:37 系统逐字引用了这条正确条目，随即判它「可能也是错的」；22:36:47 据此确定方案；22:36:47–22:37:33 完成改写；22:37:33 在用户直接纠正后反转，22:37:39 找到「副手定名莉莉安」的出处并回改

· CMML 阶段：第 4 阶段失效（评估错误）。三个阶段分别成立：

· 3 Presented：成立。该条目在注入层，系统两次逐字引用（22:36:37、22:37:39）。

· 4 Evaluated：失效。引用之后对可信度作了否定判断，而该判断与事实不符。

· 5 Decision-influential：成立。该误判直接改变了产物（11 个文件被改写），随后又被纠正。

一个需要分开写的伴因：检索的输出处理处有一个「超过 6 条就中断」的判断（`if cnt > 6: break`，系统自有代码），使打印层被截断。但回溯显示它不是主因：正确条目本来就在注入层里，模型读了、也引用了，仍作出否定判断。

· 证据：[运行] 会话 `20260923_024146_a` 的推理轨迹（消息 id `62854 → 62857 → 62862 → 62864`，含 62855 与 62864 两次引用该条目）

· [源码] 输出截断语句

· [运行] 11 个文件改写与回改的记录

· 说明：本案例原记为「检索到但未呈现」，经推理轨迹回溯改判。它与 C-12 的区别在于：C-12 是条目本身有歧义，本案例是条目正确而被误判

· 脱敏：只涉及本地模型名称与文件名

C-06 检索到但未呈现：注入层预算裁剪移出 19 条，其中 11 条无处可去

· 时间：2026-10-02

· 经过：常驻注入层触及字符预算上限，一次裁剪移出了 19 条记忆条目。裁剪时记录了「落脚点」——其中 8 条明确迁移到其他文件，11 条没有任何迁移目标，等于从可见层直接消失

· CMML 阶段：Presented 失效（内容仍在磁盘上，但不再进入上下文）

· 证据：[运行] 裁剪记录（19 条及其落脚点列）；[运行] 注入层文件在裁剪前后的 git 版本

连带：其中 3 条的迁移目标是文档目录下的文件，而该目录不在索引覆盖范围内（见 C-03），因此这 3 条既不在常驻层、也无法被检索到。

· 脱敏：无

C-07 未呈现：禁令写在不会进入上下文的层里

· 时间：规则写入 `工作纪律.md`，commit `da60afb`，2026-09-28 14:40:54

· 第一次真正的 TTS 调用在 2026-10-02 20:59:39

· 经过：禁令写在长期层的 `工作纪律.md` 里，而该文件不在常驻注入层——截至 2026-10-20 核查，`MEMORY.md` 与 `USER.md` 中 TTS / 语音相关条目为 0。写入后第一次真正的 TTS 调用发生在 2026-10-02 20:59:39（会话 `20260930_042107_b`，音色试听），而该会话的工具调用中一次也没有打开过 `工作纪律.md`

· CMML 阶段：Presented 失效（规则存在，但不在每轮可见的层）。Stored ✓；Presented ✗ —— 不是检索失败，而是这条规则从来没被放到会被读到的位置

· 证据：[运行] 注入层无 TTS 条目（`MEMORY.md` / `USER.md` 各 0 处）

· [运行] 会话 20260930_042107_b 的工具调用记录

· [源码] commit da60afb

· 说明：记录中提到的「39 次」此类违反的次数未获独立复核，本文不引用该数字

· 脱敏：只涉及本地脚本名与模型系列名

C-08 拒绝但改变计划：记下的推荐被实施、测量、否决，方案随之改变

· 时间：决策记录提交 a7dd8a5，2026-10-06 08:40:54；原始测量文件 08:34:57 与 08:36:36

· 经过：此前存入的经历记录中，推荐把排序队列换成二叉堆。智能体把这条存储的推荐当作计划执行，在同一个脚本里实现了三个变体——A（旧算法原样）、B（在块上缓存距离并内联二分查找）、C（二叉堆）——并用指纹校验做 A/B/C 对照，凡选中不同分块的变体即作废。测量结果是 C 反而更慢（20 m：A 19.24 → B 8.77，C 12.83；1000 m：B 18.78，C 24.80；10000 m：B 5.80，C 7.48，单位毫秒）。智能体否决了存储中的推荐，改用 B，并且只落地了 B

· CMML 阶段：Decision-influential 正例。存储的经验被检索进计划、被评估、因实测证据被否决，而该否决改变了实现方案（C → B）。这正是第 3.5 节把「拒绝但影响决策」纳入该阶段定义的依据

二阶细节：记录中智能体自己承认，它是先被坑了之后才去翻旧探针的——「老数没错，是我昨晚没先翻它」。这一句本身即是「写入—命中延迟」的又一个实例。

· 证据：[自述] 经历记录中的测量与结论

· [运行] 两份原始测量输出文件（cache\select35-out.txt 870 字节 / 08:34:57，select35-after.txt 868 字节 / 08:36:36）

· 复现性：生成这两份输出的脚本与基准工程已不在机器上（全盘检索 0 命中），无法重跑

· 且两次运行之间 A 组自身由 20.0 漂至 21.4 ms。因此本案例只作为单次测量记录，A/B/C 的倍数差不作为实测结论引用

· 脱敏：无个人数据

C-09 更新引入错误：同一条规则在 41 分钟内被写了两遍，随后合并

· 时间：重复写入约 2026-09-30 01:40；对等线发现约 02:0x；合并于 02:05

· 经过：注入层先写入「存储与盘位」一条；41 分钟后又写入「盘位水位线」——同一条规则的第二个版本：同样的阈值（C 盘 1%、其它盘 2.5%）、同一个看护任务、同样的「不要挪动文件」要求。对等线的自检在 41 分钟后报告了这次重复；02:05 两条被合并（保留第一条、吸收三句独有内容、删除第二条），合并前的文件保留为备份

· CMML 阶段：Updated 失效（更新本身引入冗余）

· 证据：[运行] 合并前备份文件中的两处原文（第 70 行与第 74 行）；[自述] 经历记录中对重复的描述

· 脱敏：无

C-10 完成未回写：动作早已发生，任务列表里没有回写，条目继续以"未完成"存在

· 时间：写入 2026-10-07 16:19（commit `27ddb70`）；验收实际完成 2026-10-26 凌晨；关单 2026-10-28

· 经过：这是一条需要使用者先挑一个允许网关掉线几分钟的窗口、再由系统验收「看门狗能否把网关拉回来」的待办

· 10-26 凌晨，两次真实掉线发生，看门狗经三次判定（跨约 40 秒）后把网关拉起，两次的 `boot` 分别为 23 与 24、均 `send_ok=True`，实际用时 25–45 秒，判据为 ≤5 分钟——验收条件已经满足。

· 但完成状态没有回写到任务本。此后该条继续以"未完成"的身份存在，被早间作业原样重述：21 天内累计 19 次，其中 10-17 起连续 11 次逐字相同。

· 10-28 核对日志才发现它早已完成，随即关单（28 项 → 27 项）。

· CMML 阶段：第 7 阶段失效（完成状态未回写）

阶段	判定	依据
1 Stored	成立	git 提交史完整
3 Presented	成立	条目一直在任务本中，被作业反复读取与重写
4 / 5 决策影响	成立	10-26 的验收动作确实被执行
7 Updated	失效	动作完成了，但"已完成"这个状态从未写回条目；条目内容从 10-17 起一字未改
6 行为有效	未测量	验收判据（≤5 分钟拉起）成立，但这是工程判据，不是记忆有效性对照

缺的那一环是"做完 → 回写"，不是"动手"。 这条待办的动作当天就顺手发生了；当时没有人把日志与这条待办对上号，验收完不回写，它就以未完成的身份继续躺在任务本里，然后每天被重述一遍。

它的形态与成因要分开写：形态上是「维持性复述」（19 次原样重写），成因是状态未回写。把形态当成因，会得出"重复导致被忽略"的结论——而本例恰恰相反，重复是因为任务列表里的状态没改。

由此产生的两条设计改动（已实施）：

1. 关单必须带证据：任何待办关单时须写明"哪份日志 / 哪个数证明它成立"，否则不许勾。这条直接针对"动手有、回头确认并关单没有"的流程缺口。

2. 脉搏带最老条目、同一句三天内不重复报：避免"未闭合项数不变"把一个长期滞留的条目埋掉。

· 证据：[运行] 任务本逐日提交史（19 次重写、11 次逐字一致、10-15 与 10-23 两日无提交）

· [运行] commit 27ddb70（首次写入）

· [运行] 2026-10-26 两次真事故的看门狗日志（boot=23 / 24、send_ok=True、25–45 秒）

· [运行] 压缩存档 1007-1857-第 1 次-20261007.md（用于核对 10-07 15:34 前后的实际事件——该时间点与本待办无关）

· 说明：此前本案例曾两次改写。第一次记为「写入后 7 天 13 小时未生效」，该数值基于错误的时间点，已撤回；第二次记为「内容零变化导致被忽略」，在确认 10-26 已完成验收后，该解释不成立，现改为「完成未回写」

· 脱敏：无

C-11 层错位：同一偏好两次违反之后才被归位，两天后又被移出

· 时间：规则首次写入按需层 2026-09-16 18:04（commit eb2cb70）

· 两次违反 2026-09-26 02:55:51 与 21:19:00

· 写入注入层 2026-09-26 21:23:50（commit ce085c1）

· 经过：「别用加粗当重点线」这条偏好从 09-16 起只存在于 跨会话状态.md（按需层）。09-26 同一天被用户指出两次，间隔 18 小时。第二次指出后 4 分 50 秒，该条才被写进 MEMORY.md（注入层）。此后又在 09-28 22:18 因注入层整理被移出，09-29 04:22 落进 说话与文风.md——而该文件按 README.md 的标注不进每轮注入

· CMML 阶段：Presented 失效（层错位）。规则存在且可检索，只是不在每轮可见的层

· 证据：[运行] git 提交 eb2cb70 / ce085c1 / 447e33e / 8b57716

· [运行] 会话 20260923_024146_a 中两条用户纠正

· [源码] README.md 对 说话与文风.md 的注入标注

· 说明：该规则在 13 天内三次进出注入层，两次有记录的违反都落在"不在注入层"的窗口内。写入注入层那次，其条文自带「已点过两次」——说明它是事后补写的

· 脱敏：无

C-12 内容缺陷：一条记忆把两个主体混成一句，导致选错执行入口

· 时间：2026-10-19 01:43–01:45，会话 `20261007_143607_c`

· 经过：用户要求读取一个中文论坛链接。系统按常驻规则去开浏览器（规则本身被正确执行），但选择了 Edge。记忆原文是「本机没装正常版 Google Chrome …… 真浏览器只有 Edge」——这句把"系统里装的浏览器"与"浏览器工具驱动的 Chrome for Testing"混成了一件事。一分钟后，系统在推理中自行解开歧义（「'你的 Chrome' = the Hermes browser (Chrome for Testing)」），随即认错

· CMML 阶段：内容缺陷。规则被执行了，位置也对；失效出在条目的措辞

· 证据：[运行] 会话 `20261007_143607_c` 中 01:43 与 01:44 两条推理原文

· [运行] 用户三条指认（10-12 04:22、10-12 04:25、10-19 01:43）

· [运行] 该条记忆在 2026-10-28 已按用户口径改写

· 说明：本案例原被归入「在场未执行」，经推理回溯改判为内容缺陷——因此「在场未执行」目前无已确认案例

· 脱敏：只涉及浏览器名称

C-13 未写入：口头授权缺失，失效后 27 分钟才被记录

· 时间：2026-09-16 23:40:41（用户指出）；2026-09-17 00:07:52（写入，commit `8408aeb`）

· 经过：用户此前给过「下次你自己点」的口头授权，但系统里没有对应条目，于是在下一次鼠标操作时又触发了一次不必要的询问。该约定在失效后 27 分钟被写入 `跨会话状态.md`，09-22 又更新一次（前台点击也放开）

· CMML 阶段：未写入（当时不在任何层）

· 证据：[运行] 会话 `20260914_162127_d` 中用户给出授权的原话；[运行] `跨会话状态.md:287` 与提交时间

· 说明：缺失的记忆造成了可观察的行为差异——多了一次本不必要的审批询问。这是"未写入"这一类目前唯一有行为后果记录的案例，但仍属部分确认（无注入日志可证明当时上下文里确实没有该条目）

· 脱敏：无

C-14 容量锚点：注入层上限是用户可感知的

· 时间：2026-09-19 03:52:47（用户陈述）；2026-09-27 00:55（配置上调，commit `8a6a6c7`）

· 经过：用户主动说出「我知道你核心记忆只有 2200 字左右」，说明注入层的字符上限是用户可感知的约束，而非纯内部参数。八天后该上限由 2200 提到 6400（USER 由 1375 提到 3200）

· CMML 阶段：不属失效，作为第 3 阶段的容量锚点使用

· 证据：[运行] 会话 20260914_162127_d 中用户陈述核心记忆上限的原话

· [运行] 重启清单.md:20 与提交 8a6a6c7

· 脱敏：无

C-15 层错位：清单存在，但没有触发去看它

· 时间：2026-09-26 03:04:39，会话 20260923_024146_a

· 经过：用户指出「你看你又忘了看任务列表了」。任务.md 当时已存在，且头部写有执行规矩；但「要不要先看一眼任务清单」这件事没有任何硬规矩，它只靠两处间接提示：memories\README.md 的索引表（写着"新会话先看这条"）与每轮注入的脉搏（"任务本未闭合 N 项"）。也就是说，它依赖"正好打开那个文件"

· CMML 阶段：呈现失效（层错位）与检索未触发两种读法并存，五名外部判定者在此为 2:2 平票，规则边界未闭合（见 4.4.1、4.4.2 节）

· 证据：[运行] 用户原话；[运行] 任务.md 与 README.md 的相应行；[运行] 检索"任务列表"一词在记忆库正文中 0 命中

· 说明：过程性规则没有被写成可检索条目，只以操作提示的形式散落，属"通约缺失"——原则没有被抽象成通则，只落到具体处置。

· 复算附注：本案例的判定依赖"有一个间接提示通路"这一判断，而通路的强弱没有量化标准；它属于规则未定处，换一名标注者可能判为「未呈现」或「层错位」。

· 脱敏：无

本文引用这些案例时的限制

1. 样本为 1：一台机器、一名主要用户、一个长期运行智能体，不主张外部效度。

2. 会话归属不等：15 个案例中，9 个可定位到对话内会话，3 个属于非对话事件（源码缺陷、记忆编辑、脚本执行），1 个部分定位，1 个未能核实（C-02）。非对话事件的证据来自源码与提交，不来自会话记录。

3. 判定依据分层：第 1、2、3、7 阶段的判定依赖系统侧记录（git、索引文件、注入层快照、备份）；第 4、5 阶段的判定依赖模型的推理轨迹（`messages.reasoning_content`，覆盖 97.8% 的助手消息），须与下游产物互证；第 6 阶段无对照，完全不可判定。

4. 判定由作者完成，未做独立标注者一致性检验（本轮受经费与人力限制跳过）。作为替代，每个案例都给出可复算的材料：会话 id、commit 哈希、文件行号与原始输出路径，读者可以据此独立复核任一判定，分歧可以在材料层而非判断层检验。独立标注者的可执行性测试列为后续工作。

5. 自述只用于提出机制假设：系统对自身失效机制的说明（如"只往后看、不横向回查"）不证明事实发生，但可以作为假设，须由独立产物交叉验证后才成立。

6. 不构成效果证据：这些案例说明各阶段可以被观测、失效可以发生在不同层与不同维度；它们不证明任何机制能提升性能。

7. 一条历史数值已撤回：C-10 原记「写入后 7 天 13 小时未生效」，经核查该时间点与本题无关，已改为复述率口径（19/19 复述、0 次提问、21 天停滞）。

Appendix C. Workspace Evolution Research Framework　工作空间演化研究框架

本附录收录与工作空间演化有关的研究问题与长期方向。

C.1 Research Questions　研究问题

工作空间演化的主问题与子问题已在 1.3 节列为一组：F1（主问题）与 F1.1–F1.5（子问题），其可用测量方式与数据可得性见附录 G 的表 G-1。本节不再重复列举，以免出现第二套编号。

对应的研究假设是：即使核心模型保持冻结，一个具有持续反馈、记忆更新、经验提炼、技能形成与工具调整机制的工作空间，也可能使长期运行的智能体获得优于静态工作空间的表现。这是待验证的假设，不是已被证明的结论；检验它属于 F1，不属于本文已完成的观察。

C.2 Core-Workspace Co-Evolution　核心模型与工作空间的共同演化

· 条件 A：固定核心模型 + 固定工作空间，作为基线。

· 条件 B：固定核心模型 + 演化工作空间，作为当前优先研究条件。

· 条件 C：演化核心模型 + 固定工作空间，作为未来条件。

· 条件 D：核心模型与工作空间均演化，作为未来联合条件。

应尽量控制任务分布、模型版本、工具、提示、预算和评测方法，并使用多个任务实例或重复试验，报告均值与变异，而不是只挑选成功案例。

Appendix D. Operational Method in Detail　运行方法细节

本附录收录端到端流程、上下文与会话处理、检索与注入、运行时执行与分层监督、反馈与错误处理等操作层内容。正文只保留与评估设计直接相关的部分。

D.1 End-to-end Workflow　端到端流程

完整的端到端流程可以表述为：外部环境与事件进入运行时；运行时构建上下文；上下文与当前会话状态进入核心模型；核心模型作出判断与委派；运行时执行动作并返回反馈；反馈更新工作空间；更新后的工作空间参与后续任务。记忆检索发生在构建上下文与核心模型判断之间：由低成本辅助模型给出候选线索，由核心模型决定是否深入检索，由检索环节返回结果，结果再交给核心模型判断其适用性。

D.2 Context and Session Handling　上下文与会话处理

把“适用性由核心模型判断”作为设计主张，会立即引出一个权限与成本的张力，这一点在实际系统中比在概念层面更容易看清。张力有两面：

- 权责面：检索的触发权如果完全交给模型，模型就必须先知道“有哪些东西值得回忆”，而这本身是关于工作区状态的事实。让模型凭记忆判断“是否需要回忆”，等于要求它从不存在的观测中恢复“库里有什么”——这与要求它不看时钟说出当前时间属于同一类错误。
- 成本面：如果为了让模型知道“有什么”而把全部资源常驻注入，成本按轮次线性增长。被研究系统实测的固定背负约为每轮 93 KB（含工具模式表 42 KB 与系统提示 51 KB），即每一轮都在为“可能用不到的内容”付费；而一旦把常驻层压缩，被压掉的内容就只能依赖检索。

本文建议的划分是：运行时负责提供确定事实与低成本线索（存在哪些资源、改动时间、当前时间、进程与资源状态），并对“是否允许执行”拥有权限；核心模型负责判断这些事实对当前问题的意义，以及是否值得深入检索。这一划分下，线索的粒度成为一个可调的实验变量：线索越粗（只给关键词），模型越可能想不到去查；线索越细（给出摘要或全文），成本越接近全量注入。

把这一张力显式写出，是为了避免一种常见的隐含假设：仿佛只要“让模型自己决定要不要回忆”，问题就自动解决了。实际上模型的决定质量取决于运行时先给了它什么，因此“提示粒度”应当是实验中的一个受控变量，而不是实现细节。第 5.2 节将其列入需要控制的变量。

除权责面与成本面之外，实测数据还给出第三面：资源优先级。在被研究系统上，语义检索的调用需要按需拉起一个本机嵌入模型，而该模型受资源仲裁管辖；当用户正在使用显卡时，仲裁会拒绝该请求并给出原因。需要明确的是，这不是故障，也不是冲突：与用户共享同一台机器的前提下，后台的记忆检索为前台的人让路，是这套系统明确采用的设计口径——框架为用户让路，而不是与用户争抢资源。一次正确的拒绝本身即一次成功的运行时行为。

这一面在概念层面容易被忽略，因为「记忆检索」通常被当作纯软件流程来设计。但在与用户共享同一台机器、且算力受限的条件下，记忆可用性同时是一个资源调度问题：检索的时机不只取决于模型是否需要，还取决于当下资源是否允许、以及用户是否在场。由此产生一个对研究有直接影响的推论：被让路的那次检索并非「未检索」，而是「延后检索」。二者在评估上必须分开——把让路窗口计入「检索失败」，会低估系统；把延后当作丢失，会误导设计。

这一推论给出三条设计要求：其一，记忆检索应当可排队、可重试，让路只是推迟而非丢弃；其二，评估阶段指标时，应当记录「因资源让路而未执行」这一独立状态，而不是把它与「执行了但没有命中」合并统计；其三，若用户可提前告知自己的占用时段，则让路应当从被动反应升级为主动错峰。

第三条在被研究系统上有对应实践：用户在一般情况下会提前告知自己的占用安排，系统据此在用户不在的时段集中安排批量工作、在用户在场的时段减少后台唤醒。这使「用户在场与否」成为一项可以在任务开始之前查询的运行时事实，而不是一个只能在冲突发生时被动感知的信号。其结果是，资源让路在多数情况下不表现为「撞上后被拒绝」，而表现为「事前就避开了」。

需要说明的是，提前标记与机械门控是互补而非替代关系：显式标记覆盖可预期的时段，自动门控覆盖突发的占用——用户临时打开一个全屏应用时，仲裁仍会拒绝并给出原因。两者结合，才使「不阻挡用户正常使用」这一目标在长期运行中稳定成立。本文据此把「用户在场状态」列为运行时应向核心模型提供的确定性事实之一，与当前时间、进程状态、资源可用性并列。

D.3 Retrieval and Context Injection　检索与上下文注入

检索触发权与适用性判断权的划分见 附录 D；本节说明注入侧的两个可观测事实。其一，注入层存在硬性字符预算，因此什么内容常驻眼前是需要显式设计的问题，而不是把所有可用材料都放进去。其二，被让路或未触发的检索应记为延后而不是失败，否则评估会把资源策略的代价误读为记忆机制的失效。

D.4 Runtime Execution and Layered Supervision　运行时执行与分层监督

长期任务可能出现重复调用、无进展、超时或资源消耗过高等问题。本文区分认知层判断与运行时执行控制：核心模型评估任务状态、信息是否充分和下一步策略；框架实施权限、时间、资源和工具执行边界，并在预先定义的条件下暂停或终止操作。这是设计主张，具体实现需要按系统风险与能力进一步定义。

分离的理由与代价见 6.3 节，此处只说明各层如何触发、阈值如何标定。

分层监督必须服从权限次序——用户 > 运行时对资源的界定 > Agent 自我设定的限制，用户始终是唯一最高权威：运行时的硬边界在一般情况下立即生效，但当它与用户当下的明确指示冲突时，让路的是运行时，不是用户。这不是效率取向，而是这套系统的定位——用户始终是唯一最高权威。监督机制因此是「运行时对 Agent 的约束」，而不是「系统对用户的约束」。

表 D-1　分层监督的干预阶梯（方法提案，尚未实现与验证）

层级	触发条件（示例）	运行时动作	对核心模型认知自主性的影响
L1 自主执行	未触发任何阈值	正常执行，记录轨迹	无
L2 检查提示	估计耗时超限、调用重复、进展信号缺失	注入一次性提示，要求自检	低（仅提示，不限制）
L3 结构化总结	L2 后仍无进展	要求模型总结已尝试方法、当前状态、剩余问题与下一步计划	低至中（改变输出结构）
L4 策略约束	重复调用同一工具或同一策略 N 次	限制重复调用，要求更换策略或终止该子目标	中（限制动作空间）
L5 硬边界终止	达到硬性时间、资源或安全边界	由运行时终止相应执行并保存状态快照	高（终止执行）

此方案是方法提案，而非已验证的控制算法。具体阈值不能直接视为普遍最优值，需按任务类型标定。需要强调的是，以下三种控制动作在实现与语义上并不等价，必须分别定义：

· 停止模型生成：仅结束当前一次生成，不改变任务流程。

· 取消任务流程：终止当前任务的执行计划，可能丢弃未持久化的中间状态。

· 终止已启动的外部工具：对已经产生外部副作用的动作，需要单独的权限、确认与恢复（或补偿）语义；若动作不可逆，则运行时必须在此之前设置确认点，而非事后终止。

此外，外部副作用工具的权限模型应包含幂等性标记（该动作重复执行是否安全）与可逆性标记（失败后能否恢复）。这两个标记决定了 L4 与 L5 是否可以安全介入：对不可逆且非幂等的动作，监督必须在执行前生效。

长期运行系统的实验有一个容易被忽视的前提：如果系统本身经常丢状态、异常退出或恢复失败，那么任何关于“工作空间演化是否带来能力提升”的结论都可能被运行时噪声污染。因此在讨论监督机制之前，必须先能区分失败的来源。

表 D-2　失败分类与判别方式（用于实验中的噪声控制）

类型	含义	典型表现	判别方式
工程失败	服务、进程、参数或接口层面的故障	服务未启动、端口未就绪、请求超时、切块参数不合理	同一请求在环境修复后重复执行即可成功
能力失败	模型本身无法完成该任务	在环境正常、参数合理的条件下仍稳定失败	更换模型或提高模型规格后成功率显著变化
调度失败	任务本身可以完成，但运行时因资源分配而未执行（不含主动让路）	并发占用、显存不足、权限拒绝等；注意与“为用户让路”区分：让路是设计口径，不是失败	存在明确的允许 / 拒绝记录，且事后重试可成功

这一区分有直接的实践后果：被研究系统早期曾出现“小模型回退率高”的现象，后续核查发现主要来自服务就绪、接口错误码与按上下文长度切料等工程因素，而不是模型能力不足。若不做区分，这类现象会被计入“能力退化”，进而污染演化实验的结论。因此本文建议把三类失败分别记录，并在报告中给出各自的比例（附录 D）。

与之一致地，运行时的“正确拒绝”应当被记为一次成功行为，而不是一次失败：当模型请求使用资源、而该资源正被用户的前台任务占用时，框架让路是预期结果，也是这套系统“用户非侵入性”口径的落实；把它计入任务失败率，会系统性低估系统表现，也会掩盖真正需要处理的调度错误。

因此，除了工程失败、能力失败、调度失败之外，记录还需要第四种状态：“延后”——既不记成功，也不记失败。把延后混入失败会污染研究变量；把它当作成功则会掩盖真实的可用性下降。这一状态在长期运行系统中的出现频率不低，因为用户对机器的使用是常态而非异常。

任务中止时，保留必要日志和状态快照可能有助于故障分析、恢复和经验整理，但也涉及隐私、敏感信息与存储成本。是否保存、保存哪些字段、如何脱敏与保留多久，应由系统策略决定，并与附录 D 的治理措施共用同一套策略。这些属于待实现与验证的设计问题。日志字段至少应能支撑第 3.5 节的阶段判定，否则生命周期无法被评估。

D.5 Feedback and Error Handling　反馈与错误处理

辅助模型可以从当前输入生成关键词、识别可能的记忆主题或提出候选记录；嵌入模型可以支持语义检索。核心模型结合当前情境评估候选信息，判断其是否适用，以及是否需要继续检索或改变计划。本文把这一分工作为设计主张提出，理由是：候选记忆是否适用于当前情境，取决于当前任务的目标、约束与风险，而这些信息集中存在于核心模型正在处理的上下文中，辅助模型通常只见到查询串与候选片段。

需要同时说明其代价：辅助模型提供候选信息不等于候选信息正确；核心模型作出适用性判断也不保证判断无误。当核心模型系统性地误判适用性时（例如把不同条件下的历史结论机械套

用），增加检索量不会改善行为，反而可能放大错误。第 5.2 节把这一类风险列为需要控制的变量，E.4 讨论其安全含义。

与此相关的一个可选机制需要单独说明：能力足够的辅助模型可以充当审核者，检查核心模型的回答是否存在遗漏、矛盾或其他问题，并提供独立意见；但审核意见不取代核心模型生成最终回答。在架构上这应当与「认知内判断」分开：前者是核心模型应具备的判断能力，后者是可选的输出质量控制。需要注意两点：辅助模型的意见不能天然视为正确，其价值需要按审核准确性、误报与漏报、额外延迟、以及核心模型采纳审核后结果是否真的改善来检验；在本地算力受限的条件下，审核的代价（本系统中一个较大参数量的本地模型可用但较慢）本身也是需要计入的变量。

不能仅凭模型口头声称「我使用了记忆」来证明记忆有效：模型可能生成形式上引用历史信息的表述，而下游决策未发生任何变化；也可能确实使用了记忆，但因工具失败导致任务仍未完成。因此评估应按 3.5 节的阶段分别记录，并保留事件级证据。

若要估计因果作用，应设计有记忆与无记忆的对照条件，或对检索结果进行消融（ablation），并尽可能保持核心模型、任务、工具与提示条件一致。仅观察到记忆与正确答案同时出现，不足以证明记忆导致了性能改善。在单条轨迹中，「行为有效」无法被确证，只能被估计；第 5 章给出相应的实验设计。

Appendix E. Related Work in Detail　相关工作的展开

本附录收录正文第 2 章移出的逐项比较，以及外部记忆与检索、经验复用两条线的展开。

E.1 Agent Architecture and Runtime　Agent 架构与运行时

ReAct 将推理轨迹与行动交替组织，使模型能够根据观察结果继续推理与行动[1]。该工作说明智能体的行为需要在模型输出与环境反馈之间形成循环。本文与之关注点不同：本文不提出新的推理—行动提示格式，而关注长期运行所需的工作空间、状态与能力接口如何被组织，以及记忆在这些循环之间如何跨会话传递。

把 Agent 的运行时从模型调用中分离出来，并为其建立独立的资源与状态管理层，已有明确的先行工作。AIOS 提出把资源与模型相关服务从 Agent 应用中隔离进一个内核，由该内核提供调度、上下文管理、记忆管理、存储管理与访问控制等基础服务，并报告了最高 2.1 倍的执行加速[5]。AgentRM 从奖励建模的角度改善智能体的泛化能力[6]。Agent libOS 则面向长期运行与能力受控的自演化智能体提出运行时基底[7]。这些工作与本文共享同一个前提：长期运行智能体的问题不能只在模型层解决。

本文与它们的差别在于研究对象的选择。上述工作以运行时机制本身为贡献，通常需要在受控环境中构造实验以证明机制收益；本文则在一个已经持续运行的个人系统上，把注意力放在“运行时提供的记忆资源是否真的进入了认知过程”这一端到端问题上，并把运行时的角色限定为事实与权限的提供者（附录 D）。相应地，本文不提出新的运行时架构，也不声称自己的划分在机制层面优于既有设计。

CoALA 从认知科学与符号人工智能的传统出发，把语言智能体描述为具有模块化记忆组件、结构化动作空间与广义决策过程的系统，并以此回顾性地组织大量既有工作、前瞻性地指出方向[8]。就「为语言智能体提供统一的概念语言」这一目标而言，本文与 CoALA 方向一致；差别在于抽象层级与用途：CoALA 给出的是智能体的认知结构分类，本文给出的是记忆生命周期的阶段判定规则与相应的实验设计。在评估工具方面，MemGym 面向长时程智能体记忆构造统一环境，并报告记忆隔离分数，以把记忆表现与推理、检索、工具使用能力解耦，从而使记忆策略之间可以排除混淆因素后比较[9]。本文的七阶段划分与该目标互补：MemGym 回答「哪种记忆策略更好」，本文试图回答「某次记忆失效发生在哪一阶段」。

E.2 Comparison and Positioning　比较与定位

Hermes Agent 是所研究系统基于的开源运行时；Lilith 是具体实践与观察对象；CMML 是本文的核心贡献。

E.2.1 "记忆生命周期"已经有两个含义

2026 年出现了一批以"记忆生命周期"为名的框架。它们与本文将这个词用于不同的事，区分如下。

用法	关心什么	代表工作	阶段划分
存续型	一条记忆该以什么形态继续存在	VitaBench[10]	保留原文 / 压缩 / 整合 / 删除，四个可主动选择的操作
安全型	记忆通路会被怎样攻击与治理	长期记忆安全综述[11]	写入 / 存储 / 检索 / 执行 / 共享传播 / 遗忘回滚，六阶段
使用型（本文）	一条已经存在的记忆能否真的影响行为	CMML	存储 / 检索 / 呈现 / 评估 / 决策影响 / 行为有效 / 更新，七阶段

三种用法不是同一件事的粗细之分，而是问的问题不同：存续型问"怎么留"，安全型问"怎么防"，本文问"为什么没起作用"。

E.2.2 它们没有分开的那四段

把上表两列的阶段并排看，会看到一个共同的空档。

· VitaBench[10] 的动作空间是 `Keep-Exact / Compress / Consolidate / Delete`，全部作用在存储侧；它的评测对象是"表示选择是否重要"，用反事实的完整时域价值衡量。它不区分"被检索出来"之后发生的事。

· 长期记忆安全综述[11] 的六阶段把"检索之后"直接接到"执行（Execute）"，在那里只问"行为是否被污染的动作改变"。检索返回、进入上下文、被判断为可信、改变计划——这四件事在它那里合为一步。

· REALM[12]提出一条两段闭环：检索 → 再巩固（G(t) —检索→ (Gq, f) —再巩固→ G(t+1)），并明确以再巩固研究[16]为灵感——本文 3.1 节引用的同一锚点。它补的是反方向：检索反馈如何反过来改变记忆拓扑。它处理"用完之后写回"，不处理"用的时候有没有生效"；闭环只有两段，不含呈现、评估、决策影响与行为有效。

· 检索—利用瓶颈诊断[13]是目前最接近的一篇。它为每个问题同时生成"带检索记忆"与"不带任何记忆"两个答案，由裁判模型对比金标准，把结果分成有益 / 有害 / 被忽略 / 中性四类，并把失败分为检索失败、利用失败与幻觉三类。它明确指出"一个准确率数字把三件事混在一起——写入是否保住了信息、检索是否把它取出来、模型是否用对了"。

它比本文强的地方要说明：它有反事实对照（同一个问题在无记忆条件下再答一次），因此可以做归因；本文没有，只能做定位。

差别在单位与粒度：它的单位是一道题，回答"这段记忆对这道题有没有帮助"；本文的单位是一条记忆条目，回答"这条记忆在自己的通路上哪一环断掉"。它把利用按结果分为四类（有没有帮上），本文把利用按环节分为四段（卡在哪）。"被忽略"那一类最接近本文的第 3→5 阶段，但它不区分"没进上下文""进了没被采信""采信了没改变计划"这三种情况。

上述工作中，只有检索—利用瓶颈诊断[13]把"利用"单独测了出来，但利用内部仍未分档。而从"在上下文里"到"改变了行为"这一段恰恰最容易断、也最难看见：内容可以在注入层里、可以被逐字引用、却仍被判断为不可信（本文 C-05），也可以因条目措辞含混导致选错动作（C-12）。这些失效在"存续型"里不可见（它不问使用），在"安全型"里也会被记成"执行阶段的一次错误"而失去层次。

关于"汇聚"的一点说明。 "检索之后仍需再分"这个方向，目前有两类方法完全不同的证据：一类来自受控基准上的因子研究（一项以 3×3 因子设计在 LoCoMo 上分离检索与利用的研究[13]），一类来自本文在一个长期运行系统上的观察记录（附录 B 的 15 个案例）。两类证据来自不同场景、不同方法、不同判据，却指向同一结论。这种汇聚比单一来源更值得相信，也说明该方向不是某一个系统的偶然。

本文不主张优先：观察记录始于 2026-09，晚于该文[13]的公开版本。本文主张的是判据层面的差别（H.2.4），不是提出时间上的先后。

E.2.3 本文的定位

由此本文的增量落在三处：

1. 在"利用"内部再分出四档：呈现（是否进入本轮输入）、评估（是否被判为可信或适用）、决策影响（是否改变了计划）、行为有效（是否改变了结果）。已有工作把"利用"当作一个量；本文认为它至少是四件不同的事。前三者各有可观测事件与判定规则，第四者明确标注为必须靠对照条件才能测。

2. 以单条记忆为诊断单位，而不是以环境或系统为单位。VitaBench 评的是环境里生命周期决策值不值得做；安全综述编的是攻击—防御的分类；本文要回答的是"这一条在哪一步断掉"，因此判据必须落到单条记录、单次注入、单次计划变化。

3. 把"看不见"本身作为发现。回溯 15 个案例后，能确认的失效集中在环节 1、2、3、7，环节 4、5 各只有一个强案例，环节 6 完全未测量。这个分布不是系统特性，而是记录装置的分布——它给后来者指出的第一件事是补哪一段的记录。

同时必须说明本文没有覆盖的两段：多主体之间的记忆共享与传播（安全型里的 Share & Propagate），以及删除与回滚（Forget & Rollback）。CMML 的"更新"只涵盖单系统内的修正，不涵盖撤回与跨主体扩散；这两段留给后续工作。

E.2.4 判断的来源与本文的产物

上一节的比较说明了彼此分工，但还没有交代笔者每一步判断的来源。

一部分是已经确立的发现。 检索到的内容与最终能否被采用是两件事，这在检索与生成的边界上已被分别测量：一项工作在同一个问题上比较有记忆与无记忆两种条件下的答案，把记忆的作用分为有益、有害、被忽略与中性四类[13]。另一项工作指出，若按"检索后成功"记录贡献，大量本会成功的案例会被记到记忆头上，因此检索之后观察到的结果并不等于单条记忆的边际贡献[14]。笔者取这两处作为归因上的警示，但它们都基于特定基准，笔者不据此认为七阶段分解已被证明必要或充分。

另一部分是可作对照的框架。 VitaBench 把记忆的存续管理简化为保留原文、压缩、整合与删除四种操作，以完整时域的反事实价值衡量其后果[10]；一套安全框架按写入、存储、检索、执行、共享传播与遗忘回滚六个阶段组织长期记忆的风险[11]；REALM 用检索与再巩固两个阶段描述记忆如何因被使用而改变[12]。三者的阶段划分与本文不同：第一种问一条记忆该以什么形

态存续，第二种问通路如何被攻击，第三种问使用如何反过来改写记忆。笔者把它们作为邻域参照，并不由"存在六阶段分类"推出使用侧的四段是既有框架的空白。

第三类是术语与类比。 第 4、5 阶段的区分接近元记忆研究中的监测与控制[15]；一项在二十个前沿模型上的测量发现，回答准确率的排名与元认知敏感度的排名大体相反[16]，可作为"答得对"与"知道自己答得对不对"可以分离的经验支持。第 7 阶段回到第 1 阶段的回环则接近再巩固，即提取之后的记忆需要重新稳定[17]。笔者借用这些词，但不以它们作为阶段划分的证据，也不认为大模型的外部文本记忆与生物记忆同构。

在上述材料之上，本文提供三组内容，其主张强度并不相同。

第一组是概念上的分解：把从保存到影响结果的过程切成七段，并把"利用"按环节再分为呈现、评估、决策影响与行为有效（3.2、3.5 节）。这是笔者的工作假设，既未被证明必要，也未被证明充分。

第二组是在 Lilith 上形成的观察与判定规则：每个阶段的判定条件与反例（3.5 节）、十五个案例的逐条判定（附录 B）、由推理轨迹回溯第 4、5 阶段的方法（4.5 节与附录 I）。这一组是当前材料上的操作化产物，其可迁移性取决于能否在别的系统上复核。

第三组是尚未执行的设计：冻结核心、演化工作空间的对照方案，以及工程、能力与调度三类噪声的分离（5.2 节）。这一组只是拟议的验证方案，不构成对本文任何主张的支持。

有一处分歧需要单独说明。笔者把"拒绝但改变计划"也算作第 5 阶段成立：一条记忆被查阅、被评估为不适用、并因此改变了计划，与它被直接采纳在效果上属于同类事件。已有的结果分类把"被忽略"与"中性"分开，但不区分"没有进入上下文""进入而未被采信""采信而未改变计划"这三种情形。笔者区分它们，理由是三者的修法不同：前者要改注入层，次者要改评估环节的记录与复核，后者要改提示与计划约束。

尚未完成的部分：本节只比较了 2026 年出现的记忆生命周期类框架；与 Agent 架构、检索增强生成、长期任务状态、技能库与运行时安全等方向的逐项比较仍未完成，中文文献的补充检索同样待做。这两项列在第 6 章的后续工作中。

表 E-1　相关工作的初步机制对比（截至 2026-10-28）

工作	记忆形式	生命周期阶段的显式建模	适用性判断主体	主要评估对象
ReAct (2023)	无持久记忆，上下文内推理—行动轨迹	无	—	任务成功率、可解释性
Reflexion (2023)	语言反馈 + 情节记忆	部分（保存与复用）	模型（自反思）	后续尝试的成功率
Voyager (2023)	可执行技能库	部分（技能写入与复用）	模型 + 环境反馈校验	探索进度、技能增长

工作	记忆形式	生命周期阶段的显式建模	适用性判断主体	主要评估对象
Generative Agents (2023)	自然语言经历记录 + 反思合成	部分（保存、检索、反思）	检索排序（相关性/时近性/重要性）	行为可信度（消融验证组件贡献）
MemGPT（arXiv 2023）	分层记忆 + 虚拟上下文管理	部分（分层移动与中断）	模型（通过函数调用）	多会话对话与文档分析表现
A-MEM (2025)	带属性与链接的记忆笔记网络	部分（写入、链接、演化）	模型（动态索引与链接）	六个基座模型上的任务表现
CoALA (2024)	模块化记忆组件（概念架构）	否（给出结构分类）	—（不针对单一机制）	既有工作的组织与方向识别
AIOS (2025)	内核管理的记忆服务（含存储管理）	否（以服务与调度为中心）	—（运行时提供，模型决定用法）	并发服务效率（最高 2.1× 加速）
Agent libOS (2026)	运行时基底中的能力与状态	否（能力受控为核心）	—（运行时控制能力边界）	长期运行与自演化的可控性
MemGym (2026)	评测环境（不规定记忆机制）	否（以记忆隔离分数把记忆与推理、检索、工具使用解耦，而非按阶段分解）	—（评测框架）	记忆策略的隔离排名
VitaBench (2026)	基准环境中的记忆条目	部分（表示形态的存续决策：保留原文 / 压缩 / 整合 / 删除）	学到的策略	生命周期决策的时域价值（headroom、horizon reversal、动作多样性）
Lin et al. (2026)	跨会话持久记忆（不规定形式）	是（六阶段：写入 / 存储 / 检索 / 执行 / 共享传播 / 遗忘回滚）	—（治理与验证机制）	各阶段的攻击面与治理要求
REALM (2026)	异质认知图（记忆单元 + 关系）	部分（两段闭环：检索 → 再巩固）	模型（自适应图检索与重组）	LoCoMo、LongMemEval 准确率
检索—利用瓶颈诊断 (2026)	记忆条目（写入策略 × 检索方法）	否（按题分类结果，不按环节分解）	模型	准确率、检索精确率、利用的四类结果
本文 (CMML)	事实记忆与会话记录互补双层	是（七阶段，含判定规则）	核心模型（辅助模型仅提供候选）	阶段间衰减：检索→呈现→评估→采纳→行为

该对比表的作用是界定本文的增量，而不是宣称本文覆盖了其他工作的目标。表中「生命周期阶段的显式建模」一列的判断依据是：该文献是否把阶段作为独立可测量的对象，而非其系统内部是否存在该环节。 各行的核验层级与尚未完成的核验项见表 E-1 与附录 E.3。

E.3 Verification Levels　核验层级

表 E-2　相关工作的核验层级（2026-10-28）

下表逐行标注该行判断目前核到哪一层。「摘要级」指依据 arXiv 摘要页与出版页核对，足以支持行内的机制归类与评估对象；「正文级」指该行的具体数值或机制细节需回到论文正文才能确认，目前尚未完成。

行	核验层级	依据
ReAct (2023)	摘要级	arXiv:2210.03629 摘要；ICLR 2023 camera-ready（v3）
Reflexion (2023)	摘要级	arXiv:2303.11366 摘要；「episodic memory buffer」原文表述
Voyager (2023)	摘要级	arXiv:2305.16291 摘要；技能库与自验证机制；发表状态仍为 arXiv 预印本
Generative Agents (2023)	正文级	arXiv:2304.03442 正文 §4.1：检索分数 = α·recency + α·importance + α·relevance，min-max 归一，α 均取 1；消融验证三组件
MemGPT (2023)	正文级	arXiv:2310.08560 正文 §2 与结论：主上下文与外部上下文（archival / recall 存储）之间由函数调用移动数据；控制流含 interrupts
A-MEM (2025)	摘要级	arXiv:2502.12110 摘要；「six foundation models」原文；NeurIPS 2025
CoALA (2024)	正文级	arXiv:2309.02427 正文 §2：按记忆类型分为 working / episodic / semantic / procedural，并给出「内部记忆 + 动作空间 + 决策过程」三概念
AIOS (2025)	摘要级	arXiv:2403.16971 摘要；「up to 2.1x faster」原文；COLM 2025
Agent libOS (2026)	摘要级	arXiv:2606.03895 摘要；三平面分离与能力控制
MemGym (2026)	摘要级	arXiv:2605.20833 摘要；「memory-isolated scores」原文
VitaBench (2026)	正文级	arXiv:2610.14731 正文 §3.2：动作空间为 Keep-Exact / Compress / Consolidate / Delete，价值按完整时域计算
记忆生命周期安全综述 (2026)	正文级	arXiv:2604.16548 正文 §3：六阶段定义，攻击链"写入→存储→检索→执行→共享→遗忘"
REALM (2026)	正文级	arXiv:2609.16053 正文 §2.2、§3.4.3：两段闭环式（13），以再巩固研究[16]为灵感
检索—利用瓶颈诊断 (2026)	正文级	arXiv:2603.02473 正文 §2.3：三个探针；利用按 Beneficial / Harmful / Ignored / Neutral 四类，含无记忆反事实答案
本文 (CMML)	—	本文

尚需补做的：(1) 表中各行的「主要评估对象」一列目前只核到摘要或摘要段，具体指标的定义与实验设置需在各文献的实验章节逐项核对，本轮未做；(2) CoALA 的记忆类型分类与本文的

阶段分解是两条正交的轴已在 H.2 的比较中说明，但尚未逐条对照其四类记忆模块与本文七阶段；(3) 表中未列 AgentRM 与 TOIS 记忆综述，二者仅在正文引用，不与本文构成同层对比。

在完成正文级核验前，本表应被读作基于摘要与出版页的暂定判断，不作"已逐篇核对原文"之声明。

一条在正文核验中确认的关键区别：类型轴 vs 阶段轴

读完 CoALA 正文后可以确认：它与本文的分解方式不在同一条轴上，因此不宜简单比较"谁更细"。

	分解依据	回答的问题	典型产物
CoALA 等认知架构类工作	记忆的类型	一个 agent 需要哪几种记忆	working / episodic / semantic / procedural 四类模块
Generative Agents、A-MEM	记忆的组织与检索	记忆如何被组织、如何被召回	记忆流 + 反思树；带链接的笔记网络
MemGPT	存储层级与上下文管理	如何用有限窗口模拟无限上下文	主上下文 / 外部上下文之间的分页移动
本文 CMML	单个记忆条目随时间经历的处理阶段	一条记忆从写下来到影响行为，中间会在哪几步断掉	七阶段 + 每阶段的判定规则与失效模式

前三种分解回答的是"系统应该有什么"，CMML 回答的是"一条既有的记忆在哪一步失去作用"。两条轴可以叠加使用：先按类型或层级决定系统结构，再用阶段分解去诊断某一条记忆为什么没生效。

这个区分比"阶段更多 / 更细"更能说明本文的增量，建议写入 H.2。此前表 E-1 只用"是否显式建模生命周期阶段"一列来区分，读起来像是在比谁的分解更细，容易招致"coarse-grained vs fine-grained 无实质差别"的质疑。

E.4 External Memory, Retrieval, and Context　外部记忆、检索与上下文

Generative Agents 以自然语言保存智能体的完整经历，随时间将记忆合成为更高层的反思，并动态检索以支持行为规划[4]。MemGPT 借鉴操作系统的分层记忆思想提出虚拟上下文管理，通过数据在快慢记忆层之间的移动提供超出上下文窗口的可用记忆，并使用中断机制管理控制流[19]。A-MEM 借鉴 Zettelkasten 方法，通过动态索引与链接构造相互关联的记忆网络，并允许新记忆的加入触发对既有记忆上下文表示的更新[20]。上述工作分别解决了记忆的保存、分层与组织问题；本文的关注点是它们共同留下的一个缺口：记忆被检索到并进入上下文之后，是否被采纳、是否改变行为，通常不被单独测量。

E.5 Experience Reuse and Workspace Evolution　经验复用与工作空间演化

Reflexion 使用语言形式的反馈与情节记忆，帮助智能体在后续尝试中调整决策，而无需直接微调基础模型参数[2]。Voyager 把探索过程中形成的可执行技能沉淀为技能库，再由自动课程与环境反馈筛选[3]。这两项工作与本文共同关注固定或黑箱模型条件下的外部经验利用。差异在于，它们主要把经验用于「改进下一次尝试」，而本文关注的是「经验在何时、以何种形式进入当前认知过程」，并把未进入或进入未被采纳视为需要单独记录的失效阶段。

具身方向暂只记一句：具身方向的相关工作在本稿中尚未系统梳理，其文献对照属于后续工作。

Appendix F. Figure and Table List　图表清单

图清单

表 F-1　图清单

两张图均由作者使用 Obsidian Canvas 绘制，源文件保存在作者的工作目录中，未随本文附上外部路径。

编号	标题	正文位置	当前状态
图 1	CMML 逻辑架构：Context、Memory、Agent（自我定位）与自我认知、Sessions、记忆数据类型、经验存在方式	3.2	已定稿
图 2	CMML 运行架构：含检索回环与轮次回环	3.3	已定稿

表清单

表 F-2　表清单

编号	标题	正文位置
表 E-1	相关工作的初步机制对比（截至 2026-10-28）	附录 E
表 2	经验条目的建议最小字段集	3.4
表 3	CMML 七阶段的可观测事件、判定规则与典型失效模式	3.5
表 4	CMML 七阶段在被研究系统上的实测锚点	3.5
表 5	研究对象的关键规模（2026-10-07 审计值；2026-10-20 复测值见附录 H）	4.1
表 6	最小可行实验设计（方法提案）	5.2
表 7	CMML 各阶段的隐私与安全风险及对应措施	6.4

编号	标题	正文位置
表 A-1	术语对照表	附录 A
表 B-1	观察案例的覆盖情况（含改判后状态）	附录 B
表 D-1	分层监督的干预阶梯（方法提案，尚未实现与验证）	附录 D
表 D-2	失败分类与判别方式（用于实验中的噪声控制）	附录 D
表 G-1	研究问题与可用测量方式及数据可得性	附录 G
表 G-2	观察案例记录模板（扩展用）	附录 G
表 H-1	量化陈述的口径、日期与复测结果	附录 H

图 1 与图 2 尚未导出；导出后把对应占位段替换为图片引用即可，图注与编号不需要改动。

Appendix G. Research Questions and Measures　研究问题与测量方式

表 G-1　研究问题与可用测量方式及数据可得性

#	研究问题	可用测量方式	当前是否具备数据
F1.1	哪些信息应当长期保留？	长期层条目的写入 / 删除 / 合并事件计数；条目被再次引用前的间隔	部分：版本控制提供提交历史（数百次），但"该不该保留"无标注
F1.2	经验应以什么形式表示？	经验条目的字段完整度（目标 / 条件 / 行动 / 结果 / 原因 / 不确定性 / 修正）；按形式分组的复用成功率	否：现有经验条目未按统一字段记录
F1.3	何时应把经验转化为技能？	技能形成事件与触发条件的对照；技能生成前后的任务成功率	否：技能文件不在版本控制内，无法取触发时点
Q1.1	如何让记忆真正影响后续行为？	第 5 阶段（Decision-influential）成立的任务比例；拒绝但改变计划的事件计数	部分：可人工判定，见附录 B 的 C-05 / C-08；无自动记录
F1.4	工作空间如何避免无控制增长？	各层体量随时间的变化；冗余 / 冲突 / 过时条目占比；注入层触顶次数	是：注入层有字符上限与裁剪记录，长期层有提交历史
F1.5	各类资源之间如何相互影响？	记忆更新与技能 / 工具变更的事件共现；因果需对照条件	否：目前只有共现，无法区分因果
F1	工作空间的变化能否带来稳定可重复的能力提升？（主问题，子问题见上）	冻结核心下的 A/B 对照（5.2 节）；阶段指标与任务结果	否：受控实验尚未开展

本表给出 F1 的子问题与 Q1.1 的测量方式，对应第 5.2 节的实验设计。"部分 / 否"两列的填写依据是 2026-10-16 的系统审计与记录；"

具备数据"仅指该指标可被计算，不表示已经检验过任何假设。

观察案例记录模板（扩展用）

表 G-2　观察案例记录模板（扩展用）

案例编号	时间/轮次	用户意图与任务目标	系统实际行为（据日志）	期望行为	差异所属阶段	涉及个人信息的脱敏方式
C-001						
C-002						
...						

填写要求：（1）「差异所属阶段」只能取 3.5 节七阶段之一，若无法判定则记为「未判定」并说明理由；（2）「系统实际行为」必须以日志或实际输出为依据，不得以事后回忆为依据；（3）涉及第三方信息时，应替换为不可逆的占位标识，并记录替换规则。建议案例总数不少于 20 条，其中至少 5 条来自「检索到但未采纳」这一类，因为该类案例是本文主张的直接证据来源。

G.2 候选指标与需控制的变量　Candidate Metrics and Controls

候选指标包括：任务成功率、错误率与完成时间；相关记忆召回率与不相关记忆比例；记忆影响决策及其对行为的可观察影响；历史经验复用率与复用失败率；写入—命中延迟；工作空间规模、冗余、冲突和过时内容比例；更新引入错误的频率与修正时间；检索成本、延迟、资源消耗；长时间运行中的稳定性与故障恢复表现。这些指标尚未产生可支持性能提升结论的系统性数据。

需要显式控制的变量包括：（1）核心模型版本与解码参数；（2）任务难度分布；（3）提示模板与注入位置；（4）检索算法的排序策略与召回条数；（5）工作空间的初始规模；（6）线索粒度档位（关键词 / 摘要 / 全文三档，见 附录 D）；（7）模型对候选记忆的适用性误判率（可通过人工标注子集估计）。第（5）与第（7）项尤为关键：前者决定 M1 与 M2 的差异是否可归因于资源分层本身，后者决定“未采纳”到底是检索失败还是判断失败。

Appendix H. Measurement Provenance　数据口径与复测

本节列出正文与附录中每一个量化陈述的取值、测量日期、测量方法与复测结果。凡测量日期早于 2026-10-20 的数值均标为历史值；复测在同一台机器上以只读方式重做，不修改被观测系统。

表 H-1　量化陈述的口径、日期与复测结果

#	指标	论文采用值	测量日期	测量方法	2026-10-20 复测	结论
1	会话数	322	2026-10-07	SELECT COUNT(*) FROM sessions	401	已增长；正文标为 10-07 历史值

#	指标	论文采用值	测量日期	测量方法	2026-10-20 复测	结论
2	消息数	217,552	2026-10-07	SELECT COUNT(*) FROM messages	227,477	已增长
3	数据库体积	约 1.3 GB	2026-10-07	文件大小	1,360.5 MB	一致
4	长期层字符数	约 52.6 万	2026-10-07	顶层 .md 文件字符计数	678,930	已增长约 15 万
5	注入层字符数	约 1.95 万（四个文件）	2026-10-07	MEMORY + USER + SOUL + PROFILE	16,021（三个文件；SOUL 已不存在）	口径已变，不可直接比较
6	语义索引	613 文件 / 12,407 块	2026-10-07 13:37	索引文件直读	613 / 12,407（自 10-07 未再重建，已 13 天）	数值一致，但已陈旧
7	技能数	193	2026-10-07	SKILL.md 计数	200	已增长
8	框架 cron 任务	22（启用 18）	2026-10-07	任务定义文件	23（启用 18）	已增长 1
9	Windows 计划任务	13	2026-10-07	Get-ScheduledTask 过滤	13	一致
10	常驻文件裁剪	15,079 → 10,665 字节	原稿未注明	原稿未注明	git 轨迹为 15,079 → 13,832（10-02 08:05–08:17）→ 12,574（12:18）→ 10,135（12:27）	原值把两次事件并成一次，已改正
11	检索工具调用次数	7 次	原稿未注明	原稿未注明	16 条记录 / 10 个不同调用时刻（09-30 19:47:24 → 10-05 04:34:03）	原值低估，已改正
12	检索结果条数	411	2026-10-16	检索记录行数	未重测	保留；口径由「调用次数」改为「结果条数」
13	去重查询串	29	2026-10-16	去重计数	未重测	保留
14	每轮固定背负	约 93 KB	2026-09-29	框架 prompt-size	未重测	保留，标为历史值

#	指标	论文采用值	测量日期	测量方法	2026-10-20 复测	结论
15	写入—命中延迟	约 8 天（已撤回）	原稿未注明	原稿未注明	原值基于错误时间点，不成立；改以「复述率 19/19、提问次数 0、停滞 21 天」描述	指标保留为设计构念，暂无可用实测值

两条计数口径必须声明

其一，用户消息的污染率。 `role='user'` 的消息行中有 22.2%（6,065 / 27,293） 并非用户发言，而是上下文压缩交接块（`[CONTEXT COMPACTION — REFERENCE ONLY]`）、网关来源标注与 cron / 系统通知。任何基于「用户消息」的计数若不排除这些块都会显著虚高——例如按关键词检索「你不是搞过」可得 203 条，其中真实用户表达极少。本文所有此类计数均已排除这些块。

其二，工具调用的统计方式。 按文本模式匹配工具名字段会命中工具定义文本与脚本正文（2026-09-28 一天可虚增至 503 行），与真实调用相差一个数量级。本文所有调用计数一律按 `tool_name` 字段精确统计。

这两条口径问题的共同点是：在日志型数据上，字段语义比字符串匹配可靠；一条记录未必是它所声称的那件事。 这与 4.4 节的判定原则一致：阶段判定以系统侧证据为准，不以模型自述为准。

Appendix I. Measurement Method and Operational Definitions　测量方法与操作定义

本附录给出正文所引各项数值的操作定义、采集方式、去重规则与可复现性判断。 与附录 H 的分工：K 记录取值与复测结果，L 记录怎么测出来的。

I.1 采集面：观测对象不止会话

来源	覆盖什么	不覆盖什么
state.db（messages 227,477 行）	对话轮次、工具调用与结果、助手推理轨迹	记忆写入动作、索引构建、脚本执行、跨线同步
memories/（git 仓库，591 次提交）	记忆文件的每次写入与修正，含时间与说明	文件被读取的次数（读取不留痕）
cache/memory-index/index.json	索引收录了哪些文件、多少块、何时构建	索引是否被查询过
运行日志与作业记录	定时作业的执行与输出	模型内部判断
cache/ 下的原始输出文件	脚本执行的原始测量结果	脚本本身（可能已不存在）

要点：state.db 只覆盖"人在跟模型说话"的部分。15 个案例中有 3 个（C-03、C-06、C-08）根本不是对话内事件，其证据来自源码、提交与缓存文件。任何只依赖会话记录的阶段判定，都会系统性漏掉第 1、7 阶段与第 2 阶段的机械层失效。

I.2 逐项操作定义

#	指标 / 对象	操作定义	采集方式	口径与去重规则	可复现性
1	会话数、消息数	数据库行计数	SELECT COUNT(*) FROM sessions / messages	无去重；含所有来源（私聊、群、CLI、定时作业）	可复算
2	用户发言数	剔除系统块后的 user 行	见 L.3 的排除清单	排除压缩交接块、网关来源标注、定时作业通知、委派回执	可复算
3	工具调用次数	tool_name 精确等于该工具名的行数	WHERE tool_name='<name>'	同一时刻可能有重复行（重试或双写），按时刻去重后为"调用时刻数"	可复算
4	注入层字符数	被每轮注入的文件的字符数之和	逐文件读取字符数	口径随文件集合变化：2026-10-07 为四个文件；2026-10-20 复测时其中一个文件已不存在，故为三个文件	可复算，但跨期不可直接比较
5	长期层字符数	记忆目录顶层 .md 文件的字符数之和	目录遍历 + 逐文件计数	只计顶层，不含压缩存档与归档	可复算
6	索引覆盖	索引文件中收录的文件数与块数	读 index.json 的 files / chunks	按路径前缀判断某目录是否被收录	可复算（索引文件在盘）
7	检索调用	检索工具的调用时刻序列	WHERE tool_name='<检索工具>'	16 条记录去重后为 10 个不同时刻；末次时刻单独记录	可复算
8	复述率	原样重写次数 ÷ 出现次数	读任务本 git 提交史，逐次比较该行文本	逐字相同才算一次"原样重写"；有任一字符变化即不计	可复算（git 完整）
9	提问次数	模型主动就某条目向用户发起的询问次数	在推理轨迹与会话中检索该条目相关的提问轮次	只计模型发起的；用户主动问起不计	可复算（需人工判读）
10	停滞天数	条目创建日至最近一次出现日	首次提交时间 → 最近提交时间	日历日，非精确时长	可复算
11	写入—命中延迟	见 L.4（原方法已撤回）	条目自身的出现史	—	原方法不可用

#	指标 / 对象	操作定义	采集方式	口径与去重规则	可复现性
12	阶段 4 / 5 判定	推理轨迹 + 下游产物互证	见 L.5	单独推理＝部分确认；推理与下游一致＝已确认；矛盾以下游为准	人工可复核
13	证据分级	六档：源码 / 运行 / 实测 / 自述 / 推断 / 假设	人工标注	自述不因补充上下文而升级；判定成立的底线是至少一条源码或运行证据	人工可复核

I.3 用户消息的排除清单

以下模式出现在 `role='user'` 行中，但并非用户发言，统计时必须排除：

· 上下文压缩交接块（以 `[CONTEXT COMPACTION` 开头，含 `REFERENCE ONLY` 字样）

· 网关来源标注（以 `Gateway message origin` 开头）

· 定时作业通知（含 `IMPORTANT: You are running as a scheduled cron job`）

· 后台进程回执（以 `[IMPORTANT: Background process` 开头）

· 委派批次回执（含 `ASYNC DELEGATION BATCH COMPLETE`）

· 跨线接头留言（以 `（接头：` 开头）

实测污染率：27,293 条 user 行中 6,065 条属上述模式，占 22.2%。

I.4 写入—命中延迟：原方法与撤回理由

原方法：取一个"该问题仍未解决"的时刻，减去写入时刻。

两个错误：

1. 把"未解决"当作"未被使用"。 一条待办可以因外部依赖合法地挂着，同时系统一直在处理它。

2. 把相邻事件误当同一件事。 实操中取到的那个时间点，其实际内容是另一件无关的事故，二者在同一分钟内发生。

正确方法：读取条目自身的出现史，逐次记录位置与内容；只有当出现"创建后长期零出现、且无外部阻塞记录"的区间，才构成延迟。若条目在区间内被反复重述但内容不变，那是维持性复述，不是延迟。

与本案例相关的另一条实现事实：抽样作业采用"内容不变即整跳压制"的闸门，逐字节相同时不调用核心模型。因此在这条通道上，零变化的条目不会到达模型——这与"延迟"是两回事：

前者是"没被呈现"，后者是"呈现了但很久没被使用"。两者必须分开记录，否则会把"没送到"算成"送到了没用"。

按正确方法重测的结果：某条目 21 天内被原样重述 19 次、零次提问、零次状态变更——原记的约八天延迟不成立，已撤回；改以复述率（19/19）、提问次数（0）、停滞天数（21）描述。

I.5 阶段 4 / 5 的回溯流程

1. 由案例记录取会话 id 与大致时刻，定位决策点前后各约十条消息。

2. 拉取该窗口内 `role='assistant'` 且推理内容非空的全部消息。

3. 判第 4 阶段：推理中是否显式提到该记忆条目并作出取舍或适用性判断。

4. 判第 5 阶段：该判断之后是否跟有下游产物变化（工具参数、文件编辑、计划调整），并排除其他起因（用户提示、工具报错、任务变化）。

5. 交叉验证：推理与下游不一致时，以下游为准。

覆盖率 97.8% 的口径：分母为 `role='assistant'` 的消息总数，分子为其中 `reasoning_content` 字段非空的行数。该比例衡量的是记录可读性——有多少轮的模型文字轨迹留存下来——不是第 4 / 5 阶段的判定成功率，也不说明轨迹内容是否足以判出阶段。排除项见 L.3，时间范围见 L.6。

一项限制：第 6 阶段（行为有效）无法用回溯得到。反事实需要一个不同的世界，单条轨迹无论记录多详细都不含"没有该记忆"的分支。回溯只能为第 6 阶段筛选候选项——推理中明确提到该记忆、且下游产物确实不同的轨迹，才值得进入对照实验。

I.6 时区与时间处理

数据库时间戳为 Unix 纪元秒（UTC）。以本地时间字符串直接比较会错开八小时。本文所有时刻一律先换算到 `+08:00` 本地时区再比较；涉及跨日统计时使用日历日而非精确时长。

I.7 复现与审计状态

以下四档描述的是"如何检查与是否需要重跑"，不是从高到低的单一质量等级。

状态	定义	本文中的实例	
可复算	给出语句即可在原地重跑，结果确定	会话数、消息数、索引覆盖、检索时刻序列、复述率	
可复核	需人工判读，但材料完整可查	阶段 4 / 5 判定、证据分级	

状态	定义	本文中的实例	
暂不可重跑（原始输出保留）	生成脚本、依赖环境或基准工程缺失，当前无法重新生成结果，但原始输出仍可供检查	C-08 的 A/B/C 测量。注："输出内部自洽"只表示两轮输出之间无明显矛盾，不等于原始测量正确	
已撤回	原方法有误，数值不成立	写入—命中延迟约八天	案例材料的脱敏处理与材料可得性见附录 B。