Agent 记忆的“下半场”：迈向自我进化与长时程 Agent

报告日期：2026.8.4　　中文版整理：Wang Hua　　英文原文：Wei-Chieh Huang 等

- 原文：*A Survey of Agent Memory in the Second Half: Towards Self-Evolving and Long-Horizon Agents*
- 作者：Wei-Chieh Huang、Weizhi Zhang、Yueqing Liang 等
- 发表：Transactions on Machine Learning Research，2026 年 7 月
- 版本：arXiv:2602.06052v4，2026 年 8 月 4 日
- 原文链接：OpenReview
- 中文版整理：Wang Hua

图1：基础模型 Agent 记忆路线图。展示 2023-2025 年发布的代表性记忆框架，并按记忆载体（内部、外部或混合）与记忆主体（用户中心或 Agent 中心）进行归类。

摘要

人工智能研究正在经历范式转变——从优先考虑模型创新和基准分数，转向强调问题定义和严格的现实世界评估。随着该领域进入“下半场”，核心挑战在于长时程、动态且用户依赖的环境中的实际效用，例如 Agentic 编程、深度研究和计算机使用，在这些场景中，基于 LLM 的 Agent 面临超越固定上下文窗口的上下文规模爆炸问题，必须持续积累、管理和选择性复用跨扩展交互的大量信息。

因此，记忆（2025年已有数百篇论文发表）成为弥合实用性差距的关键解决方案。记忆不仅仅是被动存储，它日益成为 Agent 自我进化的载体：短期记忆决定了哪些经验在执行过程中被感知、选择和抽象化，而长期记忆则将这些经验积累并固化为可复用的知识和技能，形成 Agent 从自身经验中改进并维持持续学习的循环。

在本综述中，我们沿三个维度提供基础模型 Agent 记忆的统一视角：记忆载体（内部参数化状态和外部检索增强存储）、认知机制（感觉、工作、情景、语义和程序性）和记忆主体（用户中心个性化和 Agent 中心经验）。

然后，我们分析记忆在单 Agent 和多 Agent 拓扑下的运作方式，并强调对记忆操作的学习策略，展示记忆管理本身如何成为一种可训练的、自我进化的能力，涵盖强化学习下的上下文筛选与组织、决策时的经验巩固，以及通过 Agent 运行框架（agent harness）、上下文工程和标准化工具调用协议正在形成的显式、可移植和可共享 Agent 技能的新生态。最后，我们回顾用于评估记忆效用的评估基准和指标，并概述各种开放挑战和未来方向。

1 引言

人工智能的格局现已发生根本性范式转变：从优先考虑基础模型架构和简化基准性能，转向强调问题定义和严格的现实世界评估。这标志着AI发展“上半场”的结束，在上半场中，进展主要由训练方法（Liu等人，2024a）、扩展律（He等人，2016；Achiam等人，2023；Gu等人，2025）和模型架构（Vaswani等人，2017）驱动，不断在标准化基准上取得更高分数（Wang等人，2024m；Krizhevsky等人，2012）。

在此期间，该领域收敛于扩展数据和模型规模的主导范式。该范式遵循大规模预训练（Minaee等人，2024）后接后训练过程（Ouyang等人，2022）的通用方案，以卓越的准确性解决传统基准问题。在定义良好的训练流程下，大型语言模型和 Agent 可以在MMLU（Hendrycks等人，2021b）或MATH（Hendrycks等人，2021c）等基准上达到超过90%的准确率。

因此，LLM 和 Agent 已从静态预测器（如传统机器学习模型）迅速演变为能够在各种任务和环境中进行复杂推理（Wu等人，2025f）、规划（Li等人，2025i）和工具使用（Huang等人，2025a；Zou等人，2025b）的通用 Agent。在本综述中，我们将基础模型 Agent 定义为由基础模型（如 LLM、视觉语言模型或世界模型）驱动、并增强感知、推理、规划、工具使用和记忆管理的自主或半自主系统（详见第2节）。

尽管在标准基准上展示了令人印象深刻的能力，报告性能与许多现实世界任务和环境中的实用性之间仍存在显著差距（Yu等人，2024b）。大多数评估协议在很大程度上简化了实验假设，设计了静态的、预定义的规则，并采用相对简短和孤立的任务设置（Cobbe等人，2021；Chen等人，2021；Budzianowski等人，2018）。

特别是，大多数最近的 Agent 评估基准都伴随着简短的 Agent 执行时间，没有多轮、长期交互（Wang等人，2024j；Lu等人，2025a）。因此，这些评估不再反映基础模型 Agent 在现实中的能力，因为现实中的交互本质上是长时程、长上下文且深度用户依赖的，具有高度复杂性。

随着领域向更现实的环境过渡，如具身 Agent（Li等人，2024b）、GUI自动化（Ye等人，2025a）、Agentic 编程和计算机使用、深度研究（Huang等人，2025c）、个人医疗保健（Zhan等人，2024）和人类-Agent 协作（Feng等人，2024；Zou等人，2025c），操作环境的复杂性急剧增加，使 Agent 面临异常庞大和动态的上下文。在此类环境中，静态的一次性能力是不够的。

相反，Agent 必须跨交互积累、保留和选择性复用信息。因此，记忆成为弥合理想化基准性能与现实世界实现和环境之间差距的关键自然解决方案（Zhang等人，2025q）。与记忆主要优化效率和成本的固定任务AI系统不同，基础模型 Agent 需要记忆来支持动态、长时程环境中的持续积累、选择性复用、跨会话保留和用户适应。

随着领域进入AI发展的“下半场”，焦点从改进训练配方转向解决现实中的关键效用问题（Bell等人，2025；Yao等人，2025）。如何设计基准来评估 Agent 在现实环境中的表现已成为最重要的挑战之一（Xu等人，2025c），特别是当 Agent 努力沿两个主要实证维度适应时：用户导向的个性化和任务导向的专业化。在这两个维度中，特定用户的长期历史或编码、网络搜索等垂直任务的交互上下文远远超出仅基于提示的机制所能容纳的范围。

随着来自日常交互的多会话数据或来自项目工作的累积上下文呈指数级扩展，依赖静态记忆机制是不够的。因此，记忆架构从静态的、预定义的、简单的机制（Hao等人，2023；Hu等人，2023）向自适应的、自我进化的、灵活的单位（Liu等人，2025g；f）演进，以智能地存储、加载、总结、遗忘和精炼记忆，从而为下游任务保留信息性经验。

这一转变在记忆层级上形成闭环。工作记忆决定哪些内容进入并保留在活动上下文中，其筛选与组织策略越来越多地通过端到端强化学习训练，而不再依赖人工启发式规则（Zhang等人，2025p；Yuan等人，2025a；Zhou等人，2025c；Yu等人，2025b）。

保留下来的经验形成情景记录，并通过反思和巩固提炼为语义事实与程序性技能（Yang等人，2025a；Liu等人，2025b；Ouyang等人，2025）。这些技能进一步被外化为显式、可组合、可共享的组件，由 Agent 自主归纳、修订和维护。

现代 Agent 运行框架（agent harness）再通过上下文工程和标准化工具调用协议将这些能力暴露出来（Wang等人，2025c；Anthropic，2025；Belikova等人，2026；Song等人，2026；Zhang等人，2026a）。

因此，记忆不仅仅是 Agent 历史的接口，而是 Agent 自我进化本身的载体（Gao等人，2025a），贯穿认知机制（第3.2节）、记忆操作（第4节）和学习策略（第5节）的一条主线。

尽管基础模型 Agent 记忆研究的快速增长已产生若干综述，但在现实世界 Agent 效用环境中从系统设计角度分析 Agent 记忆方面仍存在重要空白。这种记忆设计已从简短、孤立的提示转向长时程交互，Agent 必须在爆炸性上下文窗口、多会话工作流和持久的现实世界用户关系上运作。

早期工作通常按任务应用或管理策略来组织记忆（Zhang等人，2025q；Du等人，2025），或采用神经科学启发的视角，通过功能类比和记忆生命周期将AI记忆投射到人类记忆上（Wu等人，2025g；Liang等人，2025）。虽然这些方法提供了有用的概念基础，但它们没有系统地表征底层记忆载体，也没有明确建模记忆在 Agent 系统中所服务的对象，这在上下文超出基础模型限制时尤为重要。

因此，它们未能区分 Agent 记忆的优化目标，并忽视了跨互补维度的联系，而这些联系对于在现实世界应用中设计和部署自主 Agent 系统至关重要。最近的工作开始拓宽这一概念格局。特别是，Hu等人（2025d）沿形式、功能和时间动态组织 Agent 记忆。尽管提供了有价值的记忆整合，但它仍然在很大程度上是局部的，主要关注记忆如何在 Agent 中心任务中发挥作用，而非记忆应如何为使用者设计、优化和部署。

本综述的动机源于当前文献在检索增强生成系统、长上下文建模、认知记忆架构、个性化、持续学习、反思和 Agent 评估之间的碎片化，使研究人员难以理解不同记忆机制之间的关系以及如何为现实部署设计记忆系统。

我们的综述通过三个贡献填补了这一空白：（1）一个连接载体、机制和主体的三维分类法——这些维度在先前综述中通常被分开讨论；（2）对记忆如何在单 Agent 和多 Agent 拓扑中实例化、操作和评估的系统设计分析；（3）将记忆定位为AI发展“下半场”中现实世界效用和 Agent 自我进化、自我改进的核心机制，并提出六个开放挑战以指导未来研究。

为此，我们分析了数百篇论文中的基础模型 Agent 记忆，从三个主要的互补视角出发，将记忆与日益复杂环境中系统级设计选择联系起来。

具体而言，我们引入了一个统一的分类法，围绕三个核心设计维度组织：记忆载体、认知机制和记忆主体，如图2所示。我们按载体将现有基础模型 Agent 记忆分为外部和内部，按认知机制分为感觉、工作、情景、语义和程序性记忆，按主体分为用户中心和 Agent 中心。

从系统视角出发，我们进一步分析这些记忆系统如何在不同 Agent 拓扑下运作，区分单 Agent 系统中的基本记忆操作与多 Agent 设置中的记忆路由。我们也强调学习策略日益增长的作用：Agent 正被训练来掌握记忆管理本身，从交互历史中学习，并随时间自我进化。

这些学习策略在短期选择和长期巩固之间形成闭环，将记忆从被动管理的资源转变为主动进化的能力。为反映环境变化对记忆设计的影响，我们讨论其在交互时程、环境复杂度以及 Agent 数量上的扩展问题，并回顾评估方法与指标。最后，我们提出六个开放挑战。

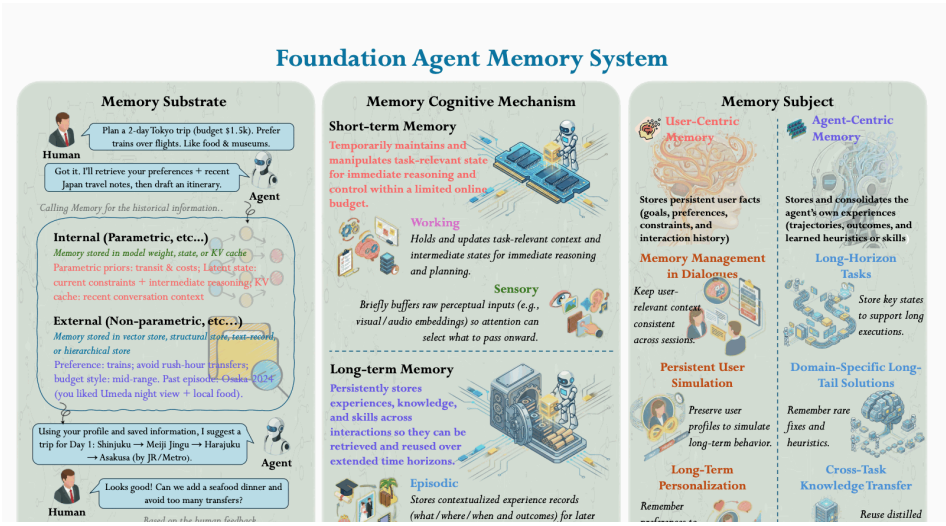

图2：基础模型 Agent 记忆的三维分类法。三个正交维度分别是记忆载体、记忆认知机制和记忆主体。

2 背景

2.1 大型语言模型与基础模型 Agent

最近的进展已将大型语言模型推出单轮问答的范畴，进入基础模型 Agent（Park等人，2023；Xi等人，2025a；Luo等人，2025；Liu等人，2025a）：能够感知环境、通过复杂目标进行推理并执行动作以实现指定目标的系统。与传统的单轮问答式聊天机器人不同，Agent 在一个完整循环中运作：它解释指令、选择动作、观察结果并更新其内部状态或记忆。这种迭代交互使 Agent 特别适合长时程、动态的问题解决。

这一趋势已在Deep Research和Manus等新兴的 Agent 产品中显现，这些产品强调多步执行和工具增强决策而非单轮响应（Zhang等人，2025l）。

基础模型 Agent 通常是一个自主或半自主系统，使用基础模型（如 LLM）作为其核心决策模块，并增强状态估计、动作执行和记忆管理机制。在本综述中，我们使用“基础模型 Agent”一词指代其核心决策由通用基础模型驱动的AI Agent，包括大型语言模型、视觉语言模型或学习到的世界模型。

核心能力通常包括任务分解和决策制定的规划（Yao等人，2023；Huang等人，2024b）、通过外部函数或模型的工具使用（Wu等人，2025c；Yuan等人，2025c；Lu等人，2025a）、多模态感知（Liu等人，2023a；Bai等人，2025；Wang等人，2025t）。

记忆（Zhang等人，2025q；Xiong等人，2025d）变得尤为重要，因为上下文窗口有限且环境随时间演化；因此，许多 Agent 依赖外部记忆存储，结合总结（Lu等人，2025b）、反思（Renze & Guven，2024）或压缩经验为可复用知识的巩固程序（Kang等人，2025c；Yu等人，2025b）。

越来越多的研究工作进一步将基础模型 Agent 框架为在部署时从其自身的执行经验中改进的自我进化系统，而非仅通过离线训练（Gao等人，2025a），并且新兴基准已开始直接评估这种通过自我进化记忆进行的经验驱动测试时学习（Wei等人，2025d）。记忆是使这种自我进化成为可能的载体：没有记录、巩固和复用经验的机制，每个事件都使 Agent 保持不变，长时程交互退化为一系列孤立的一次性任务。

多 Agent 系统（Talebirad & Nadiri，2023；Wu等人，2024b）通过为多个 Agent 分配专门角色和记忆来扩展这一范式，这些 Agent 进行通信和协调（Lan等人，2024；Estornell等人，2025）。

基础模型 Agent 现在正被探索用于不同的现实世界应用，如工作流自动化（Xiong等人，2025c）、辅导（Wang等人，2025l）、Web和GUI交互（He等人，2024a；Wang等人，2025e）、模拟或现实环境中的具身控制（Fan等人，2022；Yang等人，2025c），以及早期形式的 Agent 驱动的科学辅助（Ren等人，2025；Pantiukhin等人，2025；Zheng等人，2025d）。

尽管取得了这些进展，若干基础性挑战仍然存在。首先，长时程可靠性（Huang等人，2024b；Xi等人，2025b）需要防止累积错误、行为循环和不可靠的重新规划。其次，评估（Yehudai等人，2025）应超越静态QA，转向衡量动态、交互能力的基准，如工具使用、多步决策和长时程反馈。

第三，对齐和安全性（Hua等人，2024；Yuan等人，2024a；Tian等人，2023；Zhang等人，2024d）随着自主性和工具访问的扩展变得越来越重要，要求更强的可控性保证。解决这些挑战最终可能将今天的工具使用助手转变为足够强大以解决更复杂任务、且足够可信以在现实世界部署中保持可预测和可控的 Agent。

2.2 记忆

记忆通常指系统随时间保留、组织和利用信息的能力。在基于 LLM 的基础模型 Agent 的背景下，记忆用于解释 Agent 如何超越单轮上下文以支持长期交互、行为一致性和经验积累（Luo等人，2025）。

虽然生物学研究通常将记忆表征为持久的、经验驱动的神经变化（如突触可塑性和巩固（Hebb，2005；Bliss & Lomo，1973；Tonegawa等人，2015）），但对于 Agent 系统而言，更相关的洞见在于记忆在实践中如何被设计、实现和使用以支持不同的功能、表征和目标。

在人类认知模型中，记忆通常被理解为在不同时间尺度上组织的多个交互子系统的集合。短期记忆支持在处理过程中对信息的临时保留和操作。感觉记忆短暂缓冲原始感知输入，使下游处理成为可能（Sperling，1960），而工作记忆在严格容量限制下运作，支持在线信息操作、推理和控制（Baddeley，2020；2000）。长期记忆支持信息在较长时间内的保留，包含多个功能不同的系统。

情景记忆存储特定时间和背景下的具体经验，语义记忆积累抽象事实和概念知识（Tulving，1972；1985），程序性记忆捕捉技能、习惯和行动策略，通常通过表现而非显性回忆隐式表达（Cohen & Squire，1980；Squire，1992）。至关重要的是，这些子系统并非孤立运作。

短期系统决定了哪些经验在处理过程中被感知、选择和抽象，而长期系统积累这些经验并将其巩固为稳定的知识和技能；它们的交互，而非任何单个存储，构成了行为随经验改进的循环（Gao等人，2025a）。我们对基础模型 Agent 也采用这种系统视角：Agent 记忆被视为一个集成层级体系，其操作——从选择、保留到巩固和技能形成——本身可以被设计、优化，并越来越多地从 Agent 自身的交互中学习。

在生物系统中，记忆的功能区分最终植根于物理载体，如突触变化（Martin等人，2000）和印记（Josselyn & Tonegawa，2020）。Agent 记忆同样必须被实例化。因此，我们在第3.1节中首先通过其载体来分析 Agent 记忆，区分内部（参数化和激活）和外部（非参数化）实现。

3 基础模型 Agent 记忆分类法

我们通过 Google Scholar 检索 Agent 记忆、长期记忆、上下文管理和个性化记忆等关键词，并人工审查主要计算机科学会议与期刊的论文集。从数百篇候选论文中反复筛选，最终纳入 2023 年第一季度至 2025 年第四季度发表的 218 篇关键工作。

图3显示，相关研究在 2025 年明显加速，第四季度的增幅尤其显著；图1则把代表性框架放回时间轴中。这个趋势反映出一个越来越明确的需求：Agent 必须依靠更智能的记忆设计，才能处理复杂环境中的长时程、长上下文任务。

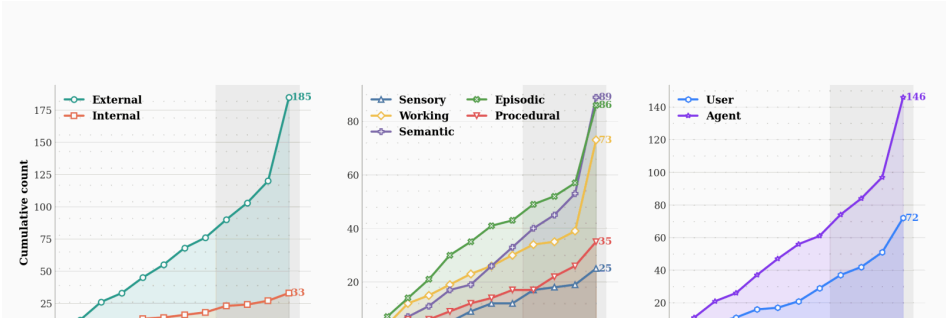

图3：2023 年第一季度至 2025 年第四季度 Agent 记忆研究的累积发表趋势。三组曲线分别按记忆载体、认知机制和记忆主体统计。

为更好地区分不同设计，我们从三个彼此正交的视角组织文献：（1）记忆载体，即记忆以何种形式存储和表示；（2）记忆认知机制，即记忆在处理流程或工作流中承担什么功能；（3）记忆主体，即记忆主要捕捉并服务谁的信息。

图4给出了完整分类树。后续章节依次讨论单 Agent 与多 Agent 系统中的记忆操作和管理（第4节）、学习策略（第5节）、扩展问题（第6节）、评估（第7节）、应用（第8节）及开放挑战（第9节）。

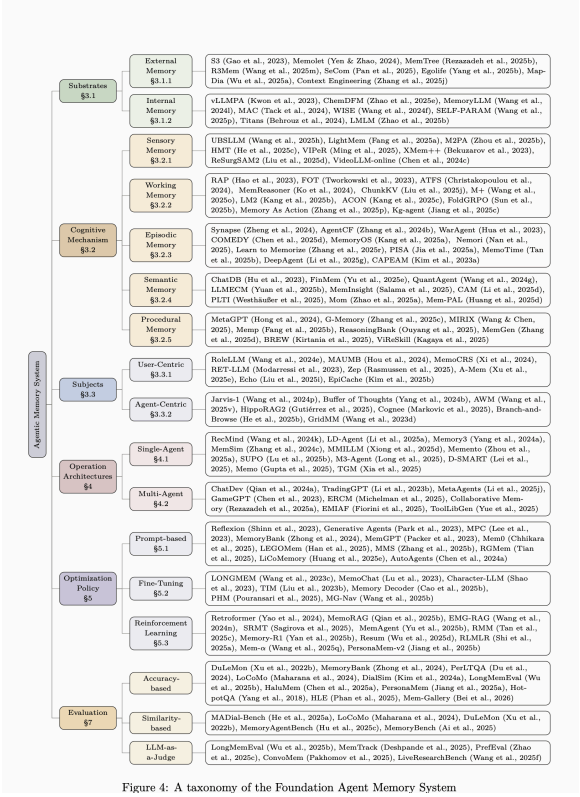

Figure 4: A taxonomy of the Foundation Agent Memory System

图4：基础模型 Agent 记忆系统的完整分类树。该图把记忆载体、认知机制、记忆主体、应用、操作机制与优化策略放在同一框架中。

3.1 记忆载体

记忆载体（memory substrate）是保存基础模型 Agent 与人类在任务和环境交互中形成的历史知识的底层机制。根据现有框架采用的存储介质和持久化方式，我们将其分为外部记忆与内部记忆，分别在第3.1.1和第3.1.2节讨论。

3.1.1 外部记忆

外部记忆指将知识、信息和过往经验存储在 Agent 模型参数或状态之外的任何记忆载体。Agent 可以通过检索和更新操作显式地从外部记忆中读取和写入，实现知识及交互历史的可扩展、易更新、跨会话保留。

外部记忆代表在向量索引、文本记录、结构存储和分层存储中存储信息的系统（Lewis等人，2020；Zhang等人，2023a）。它独立于神经网络中梯度更新的权重运作。其特点是计算（在 LLM 内部参数内执行）与知识（存储在外部数据库或记忆模块中）的清晰分离（Mallen等人，2023；Zhong等人，2024）。

这种分离的计算和检索设计允许 Agent 访问广泛且持续更新的信息，而无需昂贵的基础模型重新训练（Omidi等人，2025；Aratchige & Ilmini，2025）。此外，外部记忆既可扩展又灵活。只需少量更改，它就可以轻松地添加、替换或插入到大多数框架中。

此外，与内部记忆（Saha等人，2021）不同，知识可能因新经验的权重更新而被覆盖或调整，外部记忆可以以其原始形式保留过往信息，通过利用额外的存储模块帮助减少幻觉或知识截止问题（Castrillo等人，2025）。

然而，这种设计也有缺点和权衡。通常，推理耗时更长，因为 Agent 必须从外部存储中检索信息。当 Agent 在多轮迭代或复杂环境中运行时，此问题变得更加严重。此外，检索可能不可靠。如果相似度或排序算法不完善，它们可能提供与当前任务无关或无益的结果。这种无关信息会引入噪声，可能降低 Agent 的效用（Xu等人，2025a）。

随着系统运行时间延长、轮次增多以及记忆增长，存储和索引的不断升级的成本可能进一步降低效率和性能（Chen等人，2025e）。因此，构建良好的外部记忆通常需要严格的程序来调节记忆大小和质量，包括总结、选择性保留、遗忘、去重以及定期修剪低价值或过时项目（Du等人，2025；He等人，2024b）。

向量索引。向量索引通过将记忆项或查询嵌入到共享向量空间中，运行近似最近邻搜索以获取前k个项，并将这些片段附加到提示中作为有依据的上下文来实现（Lewis等人，2020；Zou等人，2025a）。当前 Agent 或 LLM 工作中外部记忆的主要实现是向量数据库，在RAG框架内使用（Quinn等人，2025；Li等人，2025i；Zhang等人，2025l）。此方法将记忆作为一个几何问题来运作。

通过将文本或多模态信息嵌入高维向量，它将每个记忆项表示为连续空间中的一个点。新查询也通过相同的嵌入机制转换为向量，Agent 搜索最近的点——代表最接近语义信息的数据。当前技术，如分层可导航小世界（HNSW）（Liu等人，2025h）、倒排文件（IVF）（Rege等人，2023）或乘积量化（PQ）（Huang & Huang，2024）索引，被开发和优化以在高维嵌入空间中高效找到最语义相似的文档或过往交互。

这种近似最近邻搜索检索到一小部分上下文片段，这些片段被映射回其原始形式并输入到 Agent 或 LLM 中，以将响应与相关的、先前未见过的信息整合（Guu等人，2020）。

文本记录。文本记录记忆将 Agent 的长期记忆视为一组显式的文本文档，而非嵌入或图（Zhang等人，2025q）。它被实现为持久的、人类可读的文本产物（例如，一个运行中的"核心摘要"加上情景日志），这些产物被定期总结或编辑，并在生成下一个响应时选择性地复制到提示中。一种常见实现保持一个持久的核心摘要，并用语义和情景列表加以扩充。语义列表存储离散事实，可通过插入、更新和删除操作进行修改。

相比之下，情景列表是带时间戳的事件和交互的按时间顺序的账本。在记忆更新期间，Agent 修改这些结构（Zhu等人，2025c；Wang & Chen，2025）。在响应生成期间，它提取有限的选择性相关信息和实例，然后将它们与主要摘要一起重新整合到提示中。该架构通过将记忆结构化为人类可读的摘要和列表来促进透明度和快速集成（Yue等人，2025；Park等人，2023）。然而，它需要细致的总结和修剪来维持简洁的核心摘要和可管理的列表。

结构存储。结构存储通过将记忆存储在显式模式中并通过符号查询或遍历检索它们来实现，然后将返回的记录格式化以用于 LLM 的上下文。基于拓扑设计的结构记忆架构（Jia等人，2025a；Bei等人，2025）可以分为关系表、基于图的记忆或基于树的记忆。关系表使用SQL从表中检索行记录。它们可以将事实和偏好存储为结构化记录，将短期交易和长期事件分离到不同表中，并使用连接和索引在不同表之间进行高效检索（Wang等人，2024d）。

基于图的记忆将情节或信息片段表示为知识图谱中的节点和边（Anokhin等人，2024）。知识图谱本质上是一个语义网络。它通常将信息组织为带有实体的节点和捕捉实体间关系的边。当新信息到达时，系统将语义嵌入和关键词搜索与遍历相结合，以检测边变化、解决冲突并维持有效的图结构（Jiang等人，2025c）。基于树的记忆将知识分层组织，树中每个节点存储文本内容的聚合摘要和相应的语义嵌入，更深的分支代表更具体的细节（Sarthi等人，2024）。

当新信息到达时，系统遍历树并基于嵌入相似性更新或创建节点以决定信息存储的位置。这种动态机制支持多级抽象和高效检索，以支持长对话任务（Rezazadeh等人，2025b）。结构记忆以其不同类型格式可以捕捉序列历史和关系，并支持多级抽象。该技术为 Agent 提供了对记忆如何存储、更新和查询的灵活性和控制。

分层存储。分层记忆将记忆分为多个存储单元，每个单元有自己的模式和存储策略。Agent 维护专门的记忆模块，而非将所有内容保存在一个单一的平面档案中（Rezazadeh等人，2025b；Zhai等人，2025）。

例如，Agent 可以维护用于持久角色和用户事实的核心记忆、用于时间敏感事件的情节记忆、用于抽象概念的语义记忆、用于逐步指令的程序性记忆、用于文档和媒体的资源记忆，以及用于敏感或私人信息的知识存储（Yang等人，2025a；Wang等人，2025q）。每个模块利用适合其内容的数据结构。

例如，情节记忆可能存储带有event_type、summary、details和timestamp的记录；语义记忆可按type、summary和source组织条目；程序性记忆将工作流编码为JSON序列；知识存储应用严格的访问控制来保护凭据和API密钥。

一个元记忆管理器（Wang & Chen，2025；Zhang等人，2024a）协调这些模块，将查询导向正确的存储，并利用专门的检索方法（如嵌入相似度（Xu等人，2025d）、BM25匹配（Hu等人，2025c）、字符串匹配（Alla等人，2025））为每个存储单元服务。这种多存储载体具有模块化、关注点分离和可扩展性等优势，使得设计专门的模式和检索路径更容易，同时支持跨长期交互的细致个性化。然而，协调多个记忆模块增加了系统的架构复杂性。

它可能潜在地增加查询多个存储时的延迟和额外资源成本，并需要设计良好的机制来维护数据一致性并在长时程任务的不同轮次之间同步更新（Sun & Zeng，2025；Li等人，2025k；Maragheh & Deldjoo，2025）。

3.1.2 内部记忆

内部记忆指直接存储在模型架构内的信息，涵盖嵌入其参数中的持久性知识（即参数化记忆）和推理过程中使用的工作状态。

权重。权重记忆通过将记忆写入参数来实现，因此模型后续无需检索外部上下文即可回忆信息。知识或经验通过预训练、后训练或有针对性的参数编辑直接嵌入神经网络的参数中（Mallen等人，2023；Wang等人，2025r；Ampel等人，2025；Wang等人，2024h）。由于这种知识是内部的，这些模型无需与外部存储通信即可高效稳健地回忆事实和过往事件。然而，维护和更新内部权重具有挑战性。

因此，近期研究将权重更新分为三种主要策略：持续学习、模型编辑和蒸馏。持续学习方法用新信息增量更新模型权重，同时试图避免灾难性遗忘（De Lange等人，2021）。如正则化变化以防止覆盖重要权重和应用软掩码选择性更新参数等方法，可以在识别新信息的同时很大程度上保留先前的领域知识（Serra等人，2018）。

模型编辑策略将特定事实关联视为局部化记忆并调整相应参数：一个研究方向定位决定性的中间层权重并应用秩一更新来改变单个事实关联，而另一条则通过跨多层应用小更新来扩展此思想以更新数千个记忆（Meng等人，2022；2023）。例如，一些方法表明使用增强数据进行审慎微调模型可以实现可比较的编辑性能。另一种方法插入少量新神经元以顺序纠正错误而不影响无关行为。

另一种方法添加校准记忆槽，存储纠正后的事实而不改变原始参数（Chhetri等人，2025；Luo & Specia，2024）。蒸馏技术将上下文知识压缩到参数本身中。它们可以通过训练学生复现提示教师的输出来将提示或上下文依赖的行为内化为模型参数（例如，提示到权重或上下文内蒸馏）。具体地，一些方法优化模型使其行为如同固定提示在推理时隐式存在（Cao等人，2025a；Padmanabhan等人，2023）。

虽然这种基于参数的记忆可以使推理时的回忆高效，但修改权重计算成本高昂，并可能引入干扰、覆盖或扭曲先前学到的知识（Meng等人，2023）。

隐状态。隐状态记忆通过跨步骤或跨段携带和重用中间隐藏状态来实现。即使原始标记不再在窗口中，早期激活也可以直接影响后续计算。中间激活或隐藏状态张量在信息流经层的前向传递过程中产生（Ibanez-Lissen等人，2025）。这些隐藏状态是层级表示，将固定的学习参数与当前提示相结合，形成驱动下一个标记预测的模型工作状态。与基于权重的记忆不同，状态记忆不是持久的。它通常在运行时激活期间创建，然后被使用，并在请求结束时被丢弃或重置。

因此，它支持会话内连贯性和推理，但除非导出到外部，否则不会跨会话持久化知识。关键的权衡是资源成本。环境必须不仅在内存中保存参数，还要保存中间状态，并且这种激活值随模型深度、批大小和精度而扩展。高效优化的隐状态管理可降低推理延迟和内存成本。一个代表性方向是跨层重构隐藏表示，以可控成本强化早期上下文（Dillon等人，2025）。

与重构不同，其他工作缓存隐藏状态向量或记忆标记的紧凑子集跨段，从而将有效上下文长度扩展到固定窗口之外，同时减少内存和计算成本（Dai等人，2019）。除了重用现有状态，隐状态记忆也可以被生成。记忆触发器和记忆编织器合成机器原生的潜在标记，反馈到模型中以丰富下游推理（Zhang等人，2025d）。同时，隐藏状态转换的结构化调制维持与先前上下文对齐的潜在轨迹，可减少长序列中的语义漂移（Carson & Reisizadeh，2025）。

此外，一些架构将压缩记忆集成到注意力机制中，以存储和重用先前段的键值对，从而能够以有限记忆处理极长输入（Munkhdalai等人，2024）。其他策略将隐藏状态视为快速权重，通过推理期间的小优化步骤进行更新，以在扩展上下文上精炼内部表示（Zhu等人，2025b）。

KV缓存。在基于Transformer的 LLM 中，KV缓存是一种瞬时的、推理时记忆，可加速自回归解码（Kwon等人，2023；Liu等人，2023c）。它通过在解码期间缓存每层注意力键和先前 token 的值，并在后续 token 中重用它们来实现。在自注意力中，每个 token 被投影为键和值（Ge等人，2023；Pope等人，2023）。

如果没有KV缓存，这些矩阵在每个步骤都会为此前所有 token 重新计算，导致不必要的资源浪费并减慢长序列的生成（Cai等人，2024；Feng等人，2025b）。KV缓存存储较早 token 的键和值，在每个新步骤中，仅对新生成的 token 进行计算，并从缓存中检索其余部分。该机制显著加速推理，特别是在长输出或大型深度模型上，但代价是更高的内存使用和实现复杂性（Pope等人，2023；Kwon等人，2023）。

近期研究提出了几种高影响力的KV缓存压缩方法。一种方法认识到少量 token 对注意力分数贡献最大，并动态驱逐其余部分，平衡“高贡献 token”与最近生成的 token。该方法在仅保留缓存一小部分的情况下实现了大量吞吐量提升（Zhang等人，2023b）。另一种技术识别提示观察窗口内一致的注意力模式并聚类重要特征，从而为长输入序列实现显著的存储和速度节省（Li等人，2024d）。

另一种方法观察到不同层真正需要的键值向量数量不同，而不是给每层相同的缓存大小，它按重要性对向量进行排序，并使用二分搜索在全局预算下决定每层保留多少，以便只保留信息量最大的向量（Wang等人，2024a）。这些当前研究表明KV缓存可以被高效利用以显著提升模型性能。

3.1.3 跨记忆载体的权衡

不同记忆载体具有不同的优势和缺点。它们在访问速度、可扩展性、适应性和可靠性方面存在权衡，这些在长时程任务中会迅速显现。在实践中，选择通常归结为快速、精确的内部状态与可扩展但随增长可能变得嘈杂的外部存储。

表1总结了这些在八个操作维度上的权衡：检索延迟（访问存储信息的时间成本）、存储成本（内存占用）、更新灵活性（修改存储内容的难易程度）、上下文窗口开销（对上下文窗口的额外负担）、跨会话持久性（信息是否跨会话存活）、可扩展性（容纳增长记忆的能力）、遗忘风险（修改时知识退化的敏感性）和检索精度（定位相关存储信息的准确性）。

表1：跨八个操作维度的记忆载体权衡比较分析。

维度	内部记忆（§ 3.1.2）	外部记忆（§ 3.1.1）
检索延迟	低；通过前向传播或GPU驻留缓存访问	可变；依赖于索引结构、语料库大小和检索算法
存储成本	高；随模型深度和序列长度扩展	中等；随存储条目扩展，独立于模型架构
更新灵活性	低；需要微调、模型编辑或蒸馏	高；支持显式插入、更新和删除而无需修改参数
上下文窗口开销	对隐状态和KV缓存载体显著	中等；检索到的条目必须序列化到提示中
跨会话持久性	部分；权重持久但隐状态和KV缓存被丢弃	永久；条目跨会话维护
可扩展性	受限；参数容量在训练时固定	高；存储独立于模型架构增长
遗忘风险	参数化载体高（灾难性遗忘）；KV缓存缺乏持久性	可忽略；原始条目除非显式修改否则保留
检索精度	由学习到的关联隐式决定；KV缓存会话内高	可变；对嵌入质量和排序算法敏感
代表性工作	MemoryLLM、WISE、SELF-PARAM（参数化）；vLLM、Titans、ChunkKV、EpiCache（潜在/KV）	Mem0、Zep、A-Mem、HippoRAG2、MIRIX、MemTree

内部记忆，包括编码在模型权重中的参数化记忆，通常提供高速访问和与推理的紧密集成。然而，更新昂贵且可能因频繁修改而遭受灾难性遗忘。因此它更适合稳定的通用知识而非快速变化或用户特定信息。潜在载体如隐藏激活或KV缓存快速且短暂，非常适合会话内状态。然而，它们的短暂性和随序列长度线性扩展的特性限制了容量，使其不适合跨会话保留。此外，隐状态和KV缓存载体直接占用上下文预算，带来显著的上下文窗口开销，其存储成本随模型深度、批大小和精度扩展。

外部记忆，由向量数据库或存储组成，随经验自然扩展并支持无需重新训练的灵活编辑。然而，检索增加延迟并使性能对索引和检索质量敏感。先前工作表明，存储过多或低质量项目可能注入噪声并增加定位信息性项目的难度，从而降低紧密耦合的决策制定（Liu等人，

2024b）。检索精度进一步依赖于嵌入质量、相似度度量和排序算法，使其成为外部记忆系统的关键设计考虑。为更好地解决此问题，需要显式的记忆管理机制，如修剪、总结和分层组织。

总体而言，如表1所示互补优势和局限性表明，没有单一载体能在所有设置和环境中占主导地位。有效的系统设计越来越多地采用混合记忆架构，结合外部存储以实现可扩展性和持久性，与内部载体以实现快速、集成推理。内部和外部记忆之间的选择在根本上取决于任务要求：长时间运行、用户特定的交互受益于外部记忆的持久性和更新灵活性，而快速、通用推理则有利于内部载体的低延迟和紧密集成。

3.2 记忆认知机制

人类记忆为分析基于 LLM 的 Agent 中的记忆提供了概念框架（Kim等人，2023b；Li & Li，2024；Sumers等人，2023）。认知心理学区分了多个交互系统来解释信息如何被感知、维持和重用（Baddeley，2020；Tulving，1972）。虽然文献中已提出广泛的认知记忆类型，我们专注于一组与基于 LLM 的 Agent 特别相关的五种原子认知记忆系统：感觉、工作、情景、语义和程序性记忆。

表2通过将核心功能角色映射到代表性研究方向和说明性 Agent 级实现来总结这五种记忆系统。这五个系统构成了认知记忆的最小且架构完整的分解，而其他记忆构造如自传体（Conway & Pleydell-Pearce，2000）或前瞻性记忆（Einstein & McDaniel，1990）可以被理解为建立在其上的组合或功能抽象。

因此，我们的分类法围绕这五种原子记忆类型组织，涵盖当前 Agent 架构中通常显式或隐式实现的短期和长期记忆机制。除了各自的功能外，这五个系统在 Agent 自我进化中扮演互补角色：短期系统（感觉和工作记忆）决定哪些经验在执行中被感知、选择和抽象，而长期系统（情景、语义和程序性记忆）积累这些经验并将其巩固为可复用的知识和技能，共同形成 Agent 从自身经验中改进的循环（Gao等人，2025a）。

3.2.1 感觉记忆

感觉记忆指传入感知信号的临时保留，允许注意力和选择机制在更高级处理发生之前运作，通过短暂保持原始输入足够长的时间以使系统决定接下来关注什么。

在当前的基础模型 Agent 中，感觉记忆通常没有被显式建模。这在很大程度上是由于文本输入的高度抽象特性，其中感知处理已被折叠为符号或语言表示。与编码稳定知识的记忆系统相比，感觉记忆作为感知和认知之间的瞬时接口运作，在极短时间尺度上和跨多种感觉模态（Atkinson & Shiffrin，1968）。然而，在多模态或具身 Agent 中，类似机制以短时感知缓冲区的形式出现，如视觉、听觉或交互嵌入的缓存。

这些缓冲区通过临时保留最少处理的观察结果，在它们被过滤、总结或路由到工作记忆进行下游推理和控制之前，起到感觉阶段的作用。

只有有限数量的 LLM Agent 工作将感觉记忆显式实例化为其记忆架构中的一个独特阶段。分层和多阶段设计如HMT（He等人，2025c）、LightMem（Fang等人，2025a）和M2PA（Zhou等人，2025b）将感觉记忆建模为初始缓冲区，在向下游记忆组件选择、压缩或巩固之前保留最近的、最少处理的输入。除了这些显式形式化之外，隐式感觉记忆实现在流式和具身 Agent 中更为常见。

诸如SAM2（Ravi等人，2024）和ReSurgSAM2（Liu等人，2025d）等系统维持最近视频帧的短期感知队列，而ReKV（Di等人，2025）和V-Rex（Kim等人，2025a）依赖流式KV缓存或记忆队列在在线推理期间保留最近的 token 或视觉表示。尽管这些机制很少被标记为感觉记忆，它们通过缓冲近期观察以支持感知连续性和计算效率来发挥类似功能，并且通常与工作和语义记忆紧密耦合，而非作为独立认知模块实现。

表2：五种认知记忆系统的功能角色、核心研究方向与 Agent 级实现示例。

认知类型	功能角色	基于 LLM 的 Agent 中的核心研究方向（代表性工作）	实现示例
短期记忆			
感觉记忆	感知到了什么。在进一步处理前，短暂保留最近的视觉、听觉或其他感觉输入。	面向多模态流的感知缓冲与轻量缓存，例如短时嵌入队列和感知状态缓冲区（He等人，2025c；Fang等人，2025a；Zhou等人，2025b；Di等人，2025）。通过时间门控和选择机制稳定嘈杂或高带宽观察，供后续推理与控制使用（Mon-Williams等人，2025；Bjorck等人，2025；Black等人，2025；Ravi等人，2024；Liu等人，2025d）。	保留最近 2-5 秒音频、视频帧或传感器嵌入的滚动缓冲区，用于平滑感知并处理短暂遮挡。
工作记忆	当前正在处理什么。临时保存并操作当前信息。	在写入活动上下文前塑造表示，通过压缩、折叠和抽象减少进入上下文的信息量（Labate等人，2025；Kang等人，2025c；Wu等人，2025d；Sun等人，2025b；Ye等人，2025b）。在固定预算下维护在线状态，并让 Agent 从自身经验中学习上下文和 KV 状态的更新、淘汰与运行时控制策略（Zhang等人，2025p；Yuan等人，2025a；Zhou等	进行中的推理状态：“目标：完善综述；前文已确立分析框架；下一轮修订应保持框架一致。”

认知类型	功能角色	基于 LLM 的 Agent 中的核心研究方向（代表性工作）	实现示例
		人，2025c；Yu等人，2025b；Zhang等人，2025j；Kim等人，2025b；Liu等人，2025j；Kwon等人，2023；Ni等人，2025）。	
长期记忆			
情景记忆	发生过什么。保存具体经验及其情境。	记录并组织事件，包括决定写入什么、如何组织事件与轨迹，以及如何形成多尺度情景（Rajesh等人，2025；Yeo等人，2025a；b；Anokhin等人，2024）。在决策时完成检索、反思与经验巩固，包括自适应触发、情景选择、保留策略，以及把情景提炼为可复用知识（Yeo等人，2025b；Latimer等人，2025；Li等人，2025f；Sarin等人，2025；Alqithami，2025；Yang等人，2025a；Liu等人，2025b）。	带有时间和情境的历史交互记录："上次你偏好两页摘要；此前方案因缺少 API 密钥而失败。"
语义记忆	知道什么。保存关于世界的概念和事实知识。	把知识归纳、组织成可复用表示，包括记忆图、模式和紧凑神经表示（Zhao等人，2025a；Jia等人，2025a；Li等人，2025d；Rasmussen等人，2025；Behrouz等人，2024；Wang等人，2024l；Pouransari等人，2025）。在推理过程中控制知识访问与可靠性，包括选择性激活、验证及分布漂移下的持续修订（Jimenez Gutierrez等人，2024；Wang等人，2025m；Rezazadeh等人，2025b；Yan等人，2025b；Wang等人，2024f；Alqithami，2025）。	通过查询检索并核验可靠性的知识库，例如项目资料、用户偏好和概念定义。
程序性记忆	如何行动。保存技能和行动模式。	从经验、工具或交互轨迹中归纳并封装可复用程序，逐步外化为可移植、可共享的技能（Hong等人，2024；Fang等人，2025b；Han等人，2025；Zhang等人，2025d；Xia等人，2025；Wang等人，2025c；Anthropic，2025；Xu & Yan，2026）。在变化的上下文和长时程任务中执行、组合和调整程序，并对技能库进行自我进化式维护（Ouyang等人，2025；Li等人，2025h；Tablan等人，2025；Terranova等人，2025；Wang & Chen，2025；Belikova等人，2026；Song等人，2026）。	作为固定例程调用的可复用工作流或工具技能，例如"搜索→阅读→提取→引用"或"使用 Sanitizer 调试"。

尽管在当前 LLM Agent 研究中显式处理有限，感觉记忆可能随着 Agent 部署到多模态、具身和机器人环境中而变得越来越重要。在此类环境中，Agent 必须在实时和内存约束下处理连续的高带宽感知流，如视频帧、音频信号和本体感觉（Black等人，2025；Bjorck等人，2025；Mon-Williams等人，2025）。

在这些具身应用中，感觉记忆通常通过感知级缓冲和门控机制实例化，如短时感知嵌入缓冲区、注意力驱动过滤和时间整合，而非显式标记的认知模块。这些设计减少冗余计算、稳定部分可观察环境，并支持从原始感知输入到工作和情景记忆的原则性巩固。因此，更显式的感觉记忆建模可能成为可扩展的具身和机器人基础模型 Agent 的关键设计考虑。

3.2.2 工作记忆

工作记忆指支持复杂任务如推理、理解和学习所需信息临时存储和主动操作的短期记忆机制，使信息能在持续操作期间被主动维持。

在基于 LLM 的 Agent 设置中，工作记忆（Baddeley，2020）的核心目标是在严格的在线容量约束下维持和操作任务相关状态，使多步推理和行动能不间断地进行。由于 LLM 本质上是无状态的，工作记忆提供了显式携带和跨交互步骤更新这种状态的机制。

除了作为被动缓冲区，工作记忆管理本身日益被视为 Agent 的一种自适应能力：最近的 Agent 不再仅依赖手工设计的截断或总结启发式，而是从其自身的交互经验中学习如何决定哪些信息应进入并保留在活动上下文中，使工作记忆成为 Agent 自我进化的直接对象（Gao等人，2025a）。

基础模型 Agent 中的工作记忆最常通过活动上下文实例化，包括提示上下文、中间推理轨迹、工具输出和推理时可访问的运行时状态如键值缓存。在此实例化中，现有方法在执行过程中干预的位置有所不同。一个研究方向关注任务相关状态在写入活动上下文之前或之时如何被表示。

通过压缩（Kang等人，2025c；Wu等人，2025d）、重构（Sun等人，2025b；Ye等人，2025b）或抽象交互历史（Labate等人，2025），这些方法旨在从源头上延迟或避免上下文饱和。具体地，Labate等人（2025）将大型中间输出替换为轻量级引用，而Kang等人（2025c）；Wu等人（2025d）；Sun等人（2025b）；Ye等人（2025b）定期总结或折叠已完成的推理段以维持紧凑的工作上下文。

另一个研究方向在工作状态已经积累后处理问题。这些方法研究如何在执行过程中在固定在线预算下持续维护、更新或驱逐任务相关状态。在系统层面，运行时机制管理诸如键值缓存和调度等状态（Kim等人，2025b；Liu等人，2025j；Kwon等人，2023；Ni等人，2025）。

在 Agent 层面，越来越多的研究超越固定启发式，转向自我进化的工作记忆管理，其中筛选与组织策略本身从 Agent 自身经验中学习：Agent 通过端到端训练（通常通过强化学习）来决策在推理循环中保留、巩固或丢弃什么（Zhang等人，2025p；Yuan等人，2025a；Zhou等人，2025c；Yu等人，2025b）。

一个互补方向将工作上下文本身视为跨任务进化的载体，要么通过持续将执行经验蒸馏为结构化的、可重用的上下文（Zhang等人，2025j；Suzgun等人，2025），要么通过将运行时记忆状态暴露给 Agent 使其能在执行过程中自主归档和恢复其自身上下文（Xu等人，2026）。

总之，基础模型 Agent 中的工作记忆作为 Agent 的在线工作空间，通过严格容量约束下的活动上下文实现。现有工作表明，维持连贯的长时程推理不仅依赖于将更长的上下文窗口作为唯一解决方案，而是取决于在执行期间选择性保留和操作任务相关状态。

这一转变将工作记忆从被动上下文缓冲区重新框定为自主管理工作空间：反映了 Agent 记忆从静态、预定义机制向自适性和自我进化设计的更广泛演变，管理工作记忆的策略越来越多地从 Agent 自身经验中学习和精炼，而非预先固定。工作记忆进一步充当自我进化循环的门户，因为短期状态必须先通过它才能被巩固为情景、语义或程序性记忆，在此阶段执行的选择和抽象决定了 Agent 后续可以从哪些经验中学习（第5节）。

随着 Agent 扩展到更长的时程和更复杂的交互设置，工作记忆的进展将因此取决于原则性的且日益自我进化的状态选择和转换，而非无限制的上下文扩展。

3.2.3 情景记忆

情景记忆指一种专门用于持久存储 Agent 跨会话交互经验的长期记忆形式。它记录在特定时间和环境背景下发生的具体事件，通常组织为交互轨迹、动作序列和相关反馈。

基础模型 Agent 中情景记忆的主要作用是跨长时间跨度保留历史交互背景和结果，使 Agent 能够在与持续交互或决策相关时参考过往经验（Tulving，1983；2002）。通过维持对具体过往经验的访问，情景记忆支持动态环境中的背景重建和跨会话连续性。这使 Agent 能够将其行为以先前观察到的情境为依据，维持跨重复交互的一致性，并恢复否则将在会话间丢失的相关上下文。

基础模型 Agent 中的情景记忆最常被实例化为一个显式的经验存储库，跨会话积累交互历史，并在需要时可由 Agent 访问。从方法论角度来看，基础模型 Agent 中情景记忆的现有工作可根据其主要关注点分为两个主导研究方向。一个研究方向关注事件记录，即跨会话的交互历史如何被组织为连贯的事件记录，以保留事件结构和情境背景。

对于跨会话交互历史，Rajesh等人（2025）强调选择性事件写入和过往交互的结构化组织，使存储什么以及如何组织事件内容变得明确。对于长视频理解，Yeo等人（2025a）通过将事件与其时间和因果关系统一起来构建事件记录，实现扩展视觉经验的连贯事件级表示。更广泛地，Yeo等人（2025b）在多个时间尺度上表示事件，使事件形成和访问能够适应不同粒度级别。Anokhin等人（2024）将情景观察与语义锚点链接，使规划期间的情景回忆成为可能。

另一个研究方向关注检索和反思，即存储的事件如何在决策时被触发、选择和利用。Yeo等人（2025b）将情景检索表述为一个自适应过程，根据查询和检索历史迭代选择记忆源和时间尺度。Latimer等人（2025）定义显式的回忆和反思操作，根据与当前推理背景的相关性检索事件，并利用它们指导后续行为。在工具使用设置中，Li等人（2025f）通过将当前执行状态与从过往轨迹诱导的结构化表示匹配来检索情景经验。

对于多会话对话，Sarin等人（2025）使用会话级上下文和用户状态线索触发情景检索，以跨会话回忆相关的情景摘要。最后，Alqithami（2025）表明固定记忆预算下的保留策略直接决定了哪些事件在长时程上保持可检索。

总体而言，基础模型 Agent 中的情景记忆研究聚焦于跨会话保留具体交互经验，并支持在决策时选择性访问这些经验。现有工作主要解决两个方法论问题：事件记录和情景检索或反思。

从自我进化视角来看，情景记忆进一步充当 Agent 能力从中增长的经验载体：积累的事件不仅被检索以指导个别决策，而且通过反思和巩固被蒸馏为语义事实和程序性技能，使 Agent 能够从其自身的执行历史中持续改进（Yang等人，2025a；Liu等人，2025b；Ouyang等人，2025）。开放问题包括 Agent 应如何定义事件边界、规范情景回忆在推理过程中的影响，以及随着情景记忆扩展管理长期保留。

3.2.4 语义记忆

语义记忆指一种专门用于存储抽象事实、一般概念和结构化知识的长期记忆形式。它为 Agent 提供随时间保持稳定且可在不同情境和目标中重用的去情境化信息。

在基础模型 Agent 架构中，语义记忆作为支持“知道什么”事实推理的稳定知识库运作（Tulving，1972）。其内容通常通过从情景记忆中积累的重复事实的蒸馏和去情境化得出。当 Agent 跨多个任务遇到相似知识模式时，来自个别经验的碎片化事实信息通过总结机制被整合为通

用概念、实体关系或属性摘要。这种语义化过程使 Agent 具备类似于百科全书或技术手册的知识载体。

通过支持直接访问已验证的概念知识而非重复检索原始交互日志，语义记忆为复杂推理和决策提供连贯可靠的事实基础。重要的是，在长时程 Agent 设置中，语义记忆不是静态的，而是随着新信息的积累而不断被访问、修订和控制。

现有语义记忆方法主要在抽象知识如何被构建和稳定化，以及如何在 Agent 跨长时程推理时被选择、验证和维护方面有所不同。一个研究方向关注知识归纳和组织，研究稳定的、去情境化的知识如何从交互、文档或外部证据中蒸馏出来，并以可被 Agent 可靠重用的形式存储。

代表性方法使用分层模式（Zhao等人，2025a；Jia等人，2025a）、记忆图（Rasmussen等人，2025；Wang等人，2024n）或条目中心语义结构（Xu等人，2025c；Li等人，2025d）组织语义知识，支持长期维护和推理时的结构化访问。

其他方法将语义知识编码到神经记忆模块（Behrouz等人，2024；Wang等人，2024l）或辅助参数（Wang等人，2025p；Pouransari等人，2025）中，强调紧凑表示和快速重用，而不依赖显式外部结构。另一个研究方向关注语义知识在 Agent 推理过程中如何随着新证据积累而被激活、验证和更新。

代表性方法将语义访问视为决策时的控制过程，其中抽象知识被选择性选择、检查适用性并应用于当前推理状态，而非通过固定的相似度查找检索（Jimenez Gutierrez等人，2024；Wang等人，2025m；Rezazadeh等人，2025b；Yan等人，2025b）。

其他工作强调通过引入偏好漂移检测（Sun等人，2025a）、保留或遗忘策略（Alqithami，2025）和持续知识编辑（Wang等人，2024f）来实现长期语义可靠性，以防止过时或不一致知识降低 Agent 行为。

总之，语义记忆通过提供稳定但可修订的知识载体来补充工作和情景记忆。它作为将碎片化经验蒸馏为稳定、可重用的规律和事实的主要场所。通过从特定事件中剥离通用知识本体，它为 Agent 提供了跨领域任务所需的常识基础。随着研究进展，语义记忆正从简单文档存储向自我进化的语义网络演进，确保 Agent 在长期运作中维持准确和一致的知识系统。

3.2.5 程序性记忆

程序性记忆指一种专门用于存储如何执行任务的长期记忆形式。它编码特定场景下的操作技能、执行策略和自动化例程。与存储事实知识的记忆不同，它将复杂动作序列抽象为可重用模式，使 Agent 能够高效且连贯地完成任务。

在基础模型 Agent 架构中，程序性记忆通常反映在执行层，其中重复的动作序列、决策规则或工作流决定了高层决策如何被执行。这种知识不是绑定于单一记忆载体，而是可以从先前执行中蒸馏、通过优化学习或跨 Agent 共享。随着执行经验的积累，短暂的动作状态可以被巩固为可重用的技能或例程，使 Agent 能够在长时程交互上更一致地执行任务（Fang等人，2025b）。

程序性记忆的实例化表现出从显式非参数化模板向隐式参数化神经策略通过多样化机制的演化。经验蒸馏和元认知控制涉及将历史轨迹转换为可重用方案；例如，From Experience to Strategy将交互蒸馏为结构化策略（Xia等人，2025），而Adapting Like Humans关注测试时错误纠正的元认知例程（Li等人，2025h）。

学习优化和策略精炼强调将技能内化为参数化权重；例如，Retroformer采用策略梯度优化来精炼 Agent 动作（Yao等人，2024），BREW通过持续训练和精炼增强任务处理（Kirtania等人，2025）。多 Agent 协调和共享实践在协作环境中建立共同操作规范，如Smarter Together、MetaGPT和MIRIX所探索的（Tablan等人，2025；Hong等人，2024；Wang & Chen，2025）。

工作流外化和自动化将复杂操作转换为显式自动化模板，如 Agent Workflow Memory和LEGOMem中所见（Wang等人，2025v；Han等人，2025）。

此外，MemGen和ReasoningBank等作品中的自我进化机制促进了推理轨迹的积累以实现长期程序性进化（Zhang等人，2025d；Ouyang等人，2025），而CoEvoSkills共同进化技能生成器与替代验证器，使技能包能在无法访问真实测试的情况下被精炼（Zhang等人，2026a）。

最后，虽然Evaluating Long-Term Memory等工作标志着对评估日益增长的兴趣（Terranova等人，2025），程序性记忆评估正开始在专用基准上标准化，衡量程序性知识的任务级效用及其跨任务、角色和模型的迁移（Li等人，2026；Belikova等人，2026）。

一个值得注意的近期发展将程序性记忆重新框定为显式的、可共享的技能。早期的技能库如Voyager表明 Agent 可以将经过验证的、可重用的程序积累为不断增长的行为库（Wang等人，2025c），并且这一想法已成熟为更广泛的生态系统，其中技能被打包为指令、代码和资源的可组合捆绑包，Agent 按需加载（Anthropic，2025；Xu & Yan，2026；Wang等人，2025u）。

这种外化将程序性记忆从私有 Agent 状态转变为可移植的、人类可读的、可维护的基础设施，可跨 Agent 和模型进行检查、共享和重用。

从自我进化视角来看，技能库本身随后成为进化的对象：Agent 通过迭代的诊断-修订循环从其自身的执行轨迹中修订技能（Belikova等人，2026），通过合并、修复或淘汰技能来自主维护库的健康以控制积累的技能级技术债务（Song等人，2026），并学习在长时程上归纳和进化哪些记忆技能（Zhang等人，2026b）。但这一趋势也伴随两个值得警惕的问题。

实证上，自我生成的技能仍不如人工设计和筛选的技能，表明完全自主的技能归纳对自我进化 Agent 仍然是一个开放挑战（Li等人，2026）；此外，随着技能作为共享产物流通，它们引入了新的安全和治理风险，发现社区贡献的技能中有相当一部分包含漏洞（Liu等人，2026）。

总之，程序性记忆通过提供可重用的行动抽象来补充语义和情景记忆。它目前正经历从显式非参数化指令检索到隐式参数化神经策略的关键转变。这种进化不仅支持技能获取，还确保长时程自主 Agent 中复杂决策的稳定和高效实施。与此同时，Agent 技能生态系统的快速兴起突显了一个互补的反趋势，即程序性知识被外化为显式的、可移植的和可共享的产物；使 Agent 能够自主归纳、进化并安全治理此类技能，而非依赖人工筛选与组织，这是自我进化 Agent 的核心挑战。

3.3 记忆主体

从自我进化和长时程视角来看，记忆主体表征了记忆主要建模和服务于谁。这一维度正交于记忆载体和认知机制，对于理解基础模型 Agent 中记忆的优化目标至关重要。图5说明了这种主体级区分如何与认知记忆机制相关联。工作记忆、程序性记忆和感觉记忆主要是 Agent 中心的，反映了它们在支持许多现实世界下游任务中的角色。语义和情景记忆在用户中心和 Agent 中心设置中都出现。

对于用户，它们编码纵向交互历史和演化偏好，而在 Agent 中心记忆中，它们支持世界建模、长时程任务执行和持续经验积累。共同地，我们发现记忆主体与认知类别交错，提供了一种超越先前综述中单一分类（Zhang等人，2025q；Du等人，2025）的组织记忆研究的互补视角。我们注意到用户中心和 Agent 中心记忆是概念取向而非互斥类别。在实践中，单个系统可以同时维护用户特定偏好和可重用的任务策略。

例如，编码助手可以同时维护用户特定的编码风格偏好（用户中心）和跨许多用户积累的可重用调试策略（Agent 中心）。近期工作如 MemSkill（Zhang等人，2026b）跨两个取向评估记忆。然而，当前基准很少要求同时具备两者，这自然导致现有研究强调其中一个。

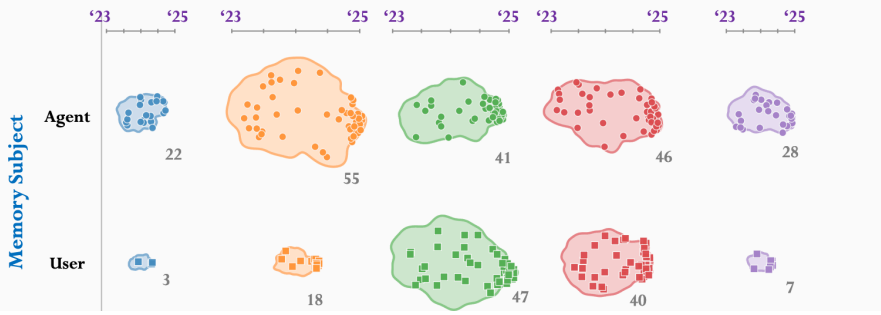

图5：记忆认知机制与记忆主体之间的关系。气泡大小表示相关研究数量；用户中心研究主要集中在情景和语义记忆，Agent 中心研究覆盖全部五类机制。

3.3.1 用户中心记忆

用户中心记忆指用户特定事实和偏好的抽象，包括传记数据、历史交互和表达偏好，这些可能跨会话和领域演化，以与用户对齐实现连贯交互和助手任务执行。

用户中心记忆构成了 LLM Agent 用于用户相关任务的基础组件，特别是长期个性化（Liu等人，2025e；Chen等人，2024b；Zhang等人，2025o）。除了仅通知临时任务执行状态的通用指令提示外，用户中心记忆通过把行动和响应建立在用户身份和跨短期交互和长期历史的演化背景中来支撑 Agent 的行动和响应（Tan等人，2025c）。

该组件在包括但不限于咨询（Litam等人，2021）、推荐（Chen等人，2018；Zhang等人，2025m；Shao等人，2026）和个人医疗保健（Vellinggiri等人，2022；Zhang等人，2026c）等领域尤为关键，Agent 必须展示对用户演化偏好、意图和行为特征的即时和纵向意识。

下面，我们不施加严格的分类法，而是强调准确和持久的用户中心记忆所服务的关键目标，使 Agent 能够持续精炼其对用户的理解并在长时程交互上维持个性化对齐。

长时程对话中的记忆管理。对话系统是用户-Agent 交互最流行的现实世界应用场景之一（Chen等人，2017；Xu等人，2025e）。在 LLM 有限的上下文窗口内，现有对话系统通常只能在有限会话上下文中处理多轮对话（Li等人，2025a）。用先前对话的所有细节提示 LLM Agent 在计算上不可行且通常适得其反，由于噪声积累和主题漂移。

有效的用户记忆因此需要原则性的相关对话上下文检索（Maharana等人，2024）、更新（Xu等人，2025e）、总结（Chhikara等人，2025；Salama等人，2025）和遗忘（Zhong等人，2024）机制。近期的现实世界记忆系统如MemGPT（Packer等人，2023）显式形式化记忆层级以管理长期对话状态。

同时，OpenAI的ChatGPT记忆功能实现了一种持久记忆机制，保留用户相关细节，同时赋予用户对存储或遗忘内容的显式控制（OpenAI，2025）。这些发展说明了对话记忆管理中的核心挑战：确定在长时程用户-Agent 交互过程中积累的哪些信息应被保留、更新或遗忘以支持未来对话。这些信息可能包括持久偏好、演化目标、重要生活事件和敏感约束。因此，记忆管理不仅是过往对话的保存，而是 Agent 随时间维持对用户连贯但可适应表征的持续过程。

持久用户模拟。高保真用户模拟器对于现实的交互在线平台至关重要，因为它们提供可扩展、可复制和受控的训练和评估环境，无需访问大量真实用户数据，从而尊重隐私并降低与实时用户研究和在线A/B测试相关的成本（Park等人，2023；Zhu等人，2024；Samuel等人，2024；Zhang等人，2025s）。

在推荐系统（Zhu等人，2025a）和社交网络（Gao等人，2023）等在线数字平台中，模拟器帮助近似长期用户行为和偏好，使得能够在动态条件下而非静态测试集上评估策略优化、排序策略和干预影响（Jin等人，2013；Zhang等人，2025k）。在此类场景中，用户中心记忆支持纵向一致性和细粒度偏好演化，因为模拟器需要维护个性化档案和跨扩展时程的自适应交互模式。

记忆启用的模拟器不应复刻固定角色，而应建模用户状态、偏好和行为如何响应积累的经验和 Agent 干预而演化。

长期个性化。与主要针对群体或平台级利益的用户模拟不同，以及主要捕捉用户-Agent 对话情境中表达偏好的对话系统（Zhang等人，2025n），长期个性化聚焦于在跨越数天、数月甚至数年的长时间跨度上优化特定用户的体验。通过跨会话维护显式用户档案或持久个人知

识库（Zhang等人，2026d），Agent 可以以持续支持同一用户随时间需求和偏好的方式适应语言风格、决策和行为。重要的是，这种个性化不应将用户档案视为静态记录。

相反，Agent 必须持续将历史证据与新观察到的交互相协调，区分持久偏好与临时状态，并在用户目标或行为变化时修订其记忆。此过程实现了自我进化的偏好对齐，其中 Agent 逐步适应其理解和行为，同时保持纵向连贯性。早期长期个性化工作（Salemi等人，2024）通过情景记忆构建生成上下文相关响应。后续研究（Tan等人，2024a）进一步探索将个人记忆编码到参数高效模块中，如基于LoRA的参数（Hu等人，2022）。

然而，这些方法要么难以充分利用丰富的个人用户数据，要么产生大量计算开销。因此，最近的框架强调能够将新观察到的行为与现有记忆结构协调以实现个性化对齐的高效记忆机制（Cai等人，2025；Zhang等人，2025o）。这一方向将个性化从一次性档案构建转向整个用户-Agent 关系中的持续的、记忆驱动的适应。

隐私保护记忆。持久用户记忆在隐私、安全和数据治理方面引入了实质性需求，因为敏感属性可能被存储、检索，并通过记忆增强 Agent 中的训练时记忆和推理时上下文泄露而无意中暴露（Mireshghallah & Li，2025）。近期工作已系统证明 LLM Agent 中的记忆模块易受定向提取攻击（Wang等人，2025a），在黑盒威胁模型下可以轻易恢复存储在 Agent 记忆中的私有用户数据。

在多 Agent 设置中，隐私风险因异构 Agent 角色和动态协作而进一步复杂化，这使得跨交互记忆库强制执行一致的隐私协议变得复杂（Shi等人，2025b）。这些风险对于自我进化 Agent 尤为重要，因为持续个性化要求记忆在长时间运行交互中被反复检索、更新、巩固并可能共享。为确保安全部署，Agent 必须支持选择性记忆保留、安全存储、用户控制的删除，以及记忆了什么或被遗忘了什么的透明审计。

它们还应进一步为用户提供对其演化档案如何被推断和更新的控制，而不仅允许控制个别存储记录。在实践中，这些保护措施通常与差分隐私机制（特别是在个性化或联邦适应中）、通过私有向量数据库的基于加密的存储和检索，以及显式保留或访问控制策略相结合，以减轻来自存储内容和嵌入的泄露风险（Tran等人，2025；Shi等人，2025b）。

3.3.2 Agent 中心记忆

Agent 中心记忆指 Agent 通过其自身的任务执行历史或通过环境交互获得的蒸馏知识、技能和操作任务先验。这支持跨现实世界环境的长上下文、长时程和长时间运行任务。

与主要保留关于相应用户的个性化信息的用户中心记忆不同，Agent 中心记忆编码来自 Agent 自身解决和与真实世界任务交互历史的更通用教训和经验（Luo等人，2025；Shinn等人，2023；Wang等人，2025c；Zhang等人，2025i）。这种记忆捕捉的教训通常是普遍适用的而非绑定于任何单个用户。

本质上，它代表了 Agent 如何“从经验和环境中学习”，保留通过先前经验获得的重要事实、策略和世界知识以改善未来性能（Wei等人，2025e；d；Zhang等人，2025i）。与优化特定用户满意度的用户中心记忆不同，Agent 中心记忆处理更广泛的真实世界问题，其中经验和解决方案预期在不同任务、环境或用户中保持有用。

这对长时程自主性至关重要：处理复杂多步任务或终身学习（Zheng等人，2025c；b）的 Agent 必须能够记住并建立在其先前遇到的基础上。除了保留经验，Agent 中心记忆为自我进化提供了基础：Agent 可以将成功和失败的轨迹蒸馏为领域知识、操作策略和可执行技能；通过后续交互精炼它们；并将可重用抽象转移到先前未见过的任务。下面，我们概述需要 Agent 中心记忆方法的关键动机和场景。

长时程任务执行。LLM Agent 经常参与需要数百或数千推理和行动步骤的任务，如编码（Jimenez等人，2024）、Web导航（Zhang等人，2025l；Zhou等人，2024）、复杂多轮决策（Shani等人，2024）或顺序工具使用（Qin等人，2024）。在这些设置中，即时工作记忆容易被积累的观察和中间推理所淹没。

Agent 中心记忆提供了一种外化机制，用于存储和检索关键中间状态，使 Agent 能够超越其原生上下文窗口运作。这种记忆通过保留子目标、动作结果、环境状态、未解决故障和临时遥远步骤之间的依赖关系直接支持正在进行的任务内的执行。因此，它使 Agent 能够恢复中断的工作流、修订早期决策，并在长时间运行交互中保持连贯进展。

例如，MEM1（Zhou等人，2025c）引入了一个端到端强化学习框架，维护紧凑的内部状态，使 Agent 能够巩固相关信息同时丢弃冗余上下文，从而在任意长的多轮交互中以近乎恒定的记忆使用运作。与MEM1互补，MemAgent（Yu等人，2025b）提出了一种为长文本处理量身定制的基于RL的记忆 Agent。它以分块方式读取长输入，并使用固定长度、可覆盖的记忆逐步更新。

这种设计使 LLM Agent 能够以线性复杂度和最小性能退化扩展到极长上下文。更近期的方法包括分层记忆模块（Hu等人，2025b）、上下文折叠方案（Sun等人，2025b）和学习到的记忆控制器（Zhang等人，2025p），它们决定存储什么以及何时压缩或丢弃过时信息。共同地，这些方法将记忆扩展到被动上下文保留之外，朝向为持续 Agent 执行进行主动状态管理。

领域特定长尾解决方案迭代。许多现实世界问题表现出长尾现象（Zhang等人，2021；Kandpal等人，2023；Park & Tuzhilin，2008），其中罕见但重要的模式、错误案例或领域特定启发式在训练数据中很少出现。Agent 中心经验记忆支持保留这些罕见见解和知识，使 Agent 能够在未来出现类似或相关案例时高效重用它们（Li等人，2024a）。

通过领域内的重复交互，Agent 可以逐步将一次性解决方案巩固为领域特定知识，并根据观察到的成功、失败和环境反馈精炼其策略。这使其能够在预训练期间最初获得的知识和行为之外实现自我进化。例如，在软件调试中（Jimenez等人，2024），大多数错误遵循常见模式，但现实世界系统通常因高度特定的配置问题、依赖冲突或环境依赖错误而失败。

类似地，在科学研究中（Ghafarollahi & Buehler，2025），虽然一般推理模式和实验程序通常在学科内共享，但许多子领域特定的实验设置（如无线通信中专门的通道编码配置）或高度领域依赖的故障排除实践（如考古学中的领域特定协议）很少跨研究人员遇到，因此不太可能在预训练数据中得到充分表示。

可比较的长尾动态也出现在复杂信息搜索（Wei等人，2025a）和专业工作流自动化（Zhang等人，2025g）中，Agent 必须处理范围狭窄、上下文依赖的问题，这些问题受益于存储定制的一次性解决方案并随时间重新应用（Yang等人，2025a）。随着这些经验积累，记忆允许 Agent 更新领域特定启发式、纠正无效策略，并在长操作时程上发展日益专业化的能力。

跨任务技能泛化。长期记忆使 Agent 能够跨任务和事件积累持久知识，支持无灾难性遗忘的持续改进（Hatalis等人，2023）。在高层面上，跨任务技能泛化关注 Agent 如何从多样化交互轨迹中抽象和保留可重用知识和技能，使泛化能够跨异构任务、领域和环境，而非在单个设置或环境中改进执行。

例如，Agent KB（Tang等人，2025b）构建了一个跨领域经验框架，将从异构 Agent 轨迹中蒸馏的高层策略和执行教训聚合到共享知识库中，使 Agent 在面对跨不同领域的新任务时能够检索和重用可迁移的问题解决知识。与特定轨迹重放不同，跨任务记忆的目标是将交互经验蒸馏为任务无关的抽象和可重用的技能原语，这些原语跨环境和目标泛化。

另一个代表性例子是ReAct（Yao等人，2023）中跨WebShop（Yao等人，2022）和ALFWorld（Shridhar等人，2021）使用的动作-思维模式。这些抽象还包括可重用的工具使用模式，如Toolformer（Schick等人，2023）和ToolLLM（Qin等人，2024）中稳定的工具使用策略，以及通过重复失败积累的错误避免执行启发式（Shinn等人，2023）。

Agent 不是记忆完整解决方案，而是可以识别重复出现的子问题，提取不变的动作模式，并重组先前学到的技能以处理新任务。这使 LLM Agent 能够逐步进化为更有能力和高效的问题解决者，表现出类似于人类跨不同任务随时间发展的专业知识的行为。

持续策略和技能进化。与跨任务泛化互补，持续策略和技能进化关注与环境交互相关的程序性记忆如何通过重复执行被创建、精炼、组合和维护。这些记忆编码如何在特定交互机制内高效执行多步动作，如Web界面、GUI系统或物理环境。例如，WebArena（Zhou等人，2024）等环境中的Web Agent（Wei等人，2025e）学习重用和精炼成功的多步浏览策略，而非从头重新探索界面。

对于GUI Agent，这种程序性记忆存储界面特定的动作序列，包括桌面或移动环境约束下的菜单遍历、小部件操作和错误恢复策略（Qin等人，2025；Wang等人，2025e）。在具身 Agent 中，程序性记忆表现为可执行技能和尊重物理动力学和动作前提条件的控制策略（Fung等人，2025；Yang等人，2025c）。这些存储的经验可以作为模板、演示或先验被重用以更高效地解决任务。

然而，自我进化 Agent 必须超越简单存储成功轨迹：它们需要从经验中归纳技能、评估其有效性、在环境变化时更新它们、在不同抽象级别分解或组合它们，并移除过时或冲突的技能。基于记忆的技能学习因此允许 Agent 在重复事件中精炼有效行为，并在动作-结果规律级别内化世界模型。随时间推移，由此产生的技能库作为进化的程序性载体，既支持熟悉领域内日益高效的执行，也支持对新任务的自适应泛化。

这种能力是长时程自主性的核心，并构成终身学习（Zheng等人，2025c）和长时间运行 Agent（Yang等人，2025a）新兴研究的基础。

4 记忆操作机制

除了记忆如何被表示以及它扮演什么认知角色之外，记忆系统由其操作所定义。图6总结了单 Agent 和多 Agent 设置中的这些操作。第4.1节涵盖单 Agent 的五个核心操作，第4.2节将其扩展到多 Agent 系统，其中架构、路由和隔离决定了经验如何在 Agent 间共享和保持一致。

4.1 单 Agent 系统的记忆操作

在单 Agent 系统中，记忆操作机制定义了基础模型 Agent 如何在长时程交互和任务执行过程中主动构建、更新、控制和利用记忆。现代 Agent 不是将记忆视为静态存储库，而是通过一系列操作来操作记忆，包括索引、检索、更新、压缩、总结、遗忘和修剪。除了管理信息，自我进化 Agent 进一步整合交互轨迹、反思执行结果，并将重复经验蒸馏为可重用的知识、策略和技能。

这些操作共同调节过往经验如何被纳入持续推理和未来决策，形成单 Agent 记忆系统的操作骨干。共同地，它们建立了一个持续适应循环，其中 Agent 检索相关经验以指导当前执行，评估由此产生的成功和失败，随后修订其记忆和技能库。这个循环使 Agent 能够在长时间运行任务中维持连贯状态，同时跨任务和交互事件逐步改进。

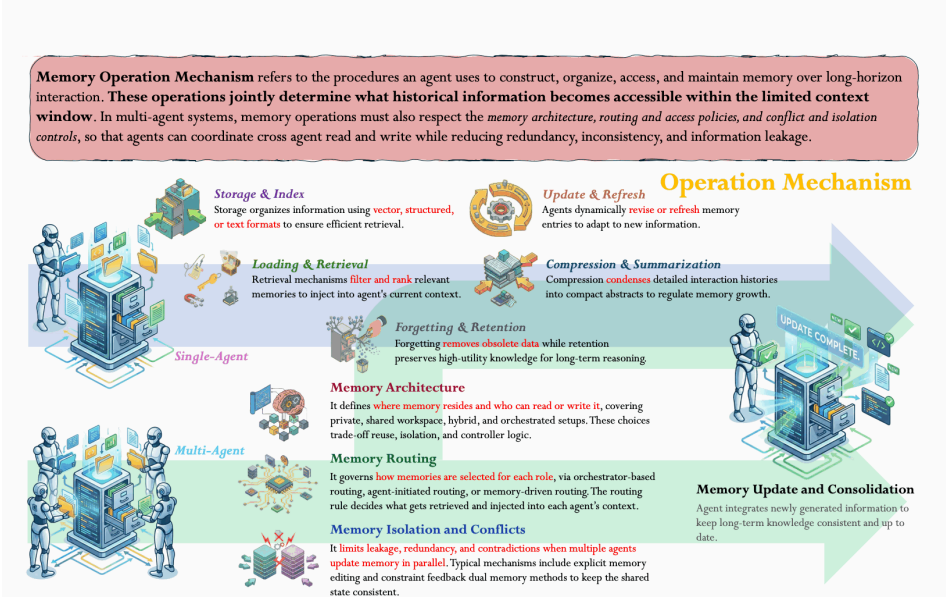

图6：基础模型 Agent 记忆系统的操作机制。单 Agent 系统包含存储与索引、加载与检索、更新与刷新、压缩与摘要、遗忘与保留五类核心操作；多 Agent 系统还需要处理架构、路由、隔离与冲突消解。

4.1.1 存储与索引

随着 Agent 记忆随时间增长，信息如何被存储和组织对于确保相关信息在需要时能被高效可靠地检索变得至关重要。在单 Agent 系统中，记忆通常在写入时通过将每个条目与语义嵌入和辅助元数据（如时间戳、任务标识符、实体或工具使用）相关联来索引（Lewis等人，2020；Zhang等人，2023a；Rezazadeh等人，2025b）。

基于向量的存储仍然是检索增强 Agent 中非参数化 Agent 记忆的主导范式，支持跨情节或语义记忆的高效近似最近邻搜索（Guu等人，2020；Quinn等人，2025）。除了平面向量索引，Agent 越来越多地采用结构化存储格式，包括关系表、基于图的记忆和分层树表示，这些支持关系查询、多级抽象和与任务结构对齐的模式感知访问（Anokhin等人，2024；Sarthi等人，2024；Rezazadeh等人，2025b）。

同时，一些系统维护文本记录记忆，存储显式的、人类可读的摘要和按时间顺序的日志，依赖关键词匹配或轻量级基于字符串的检索以优先考虑透明度和可控性（Park等人，2023；Zhang等人，2025q）。最后，存储不限于外部记忆模块：嵌入模型权重的参数化记忆、上下文窗口中的瞬时工作记忆和推理时内部状态如KV缓存共同作为影响检索行为和推理动态的隐式存储载体（Mallen等人，2023；Kwon等人，2023；Pope等人，2023）。

随着记忆跨更长时程扩展，存储格式和索引策略的选择直接影响检索精度、计算开销和下游推理可靠性。

4.1.2 加载与检索

为了在持续推理和决策过程中利用存储的经验，Agent 依赖于检索任务相关记忆，同时限制无关或过时信息的影响。在单 Agent 系统中，此过程通常从基于元数据（如新近度、任务范围或记忆类型）过滤或预选记忆条目的轻量级加载操作开始，然后是基于向量化表示的相似度检索（Lewis等人，2020；Mallen等人，2023；Zhang等人，2025q）。

加载的记忆随后使用语义相似度、启发式约束或预算感知选择策略进行排序或精炼，然后注入模型的提示或工作上下文，形成外部记忆与 LLM 推理过程之间的主要接口（Park等人，2023；Wang等人，2025q）。先前研究表明检索质量对 Agent 性能有显著影响。检索过多记忆可能引入噪声和上下文过载，而过度限制性检索可能阻止访问关键历史信息（Xu等人，2025e；Hu等人，2025c）。

因此，有效的加载和检索机制旨在平衡相关性、多样性和上下文预算，以支持连贯的长时程行为。

4.1.3 更新与刷新

随着 Agent 在扩展时程上与环境交互，先前存储的记忆可能变得不完整、过时或与新观察到的信息不一致，使得静态或仅追加的记忆表示不够充分。更新机制使单个 Agent 能够响应新观察、外部反馈或反思推理来修订现有记忆条目，允许记忆内容演化而不仅仅是积累。

在实践中，更新通常在任务完成、评估信号或检测到不一致后触发，可能涉及重写语义摘要、合并重叠的情节记录或调整存储信息的重要性（Park等人，2023；Tan等人，2025c；Wang等人，2023a）。同时，刷新操作专注于调整记忆的相对突出性和可访问性而不改变其核心内容，如重新排列显著条目、重新生成浓缩摘要或强化频繁访问的记忆以保持其对未来推理的影响。

最近的 Agent 系统进一步表明，反思或自我评估过程可以自主触发更新和刷新动作，导致长期连贯性和任务性能的改善（Shinn等人，2023；Ouyang等人，2025）。共同地，这些机制使记忆表示能够动态演化，支持非平稳环境中的适应，同时减轻过时或误导信息的积累。

4.1.4 压缩与摘要

长时程 Agent 记忆系统需要调节增长同时保留对未来推理和决策至关重要的信息的机制。压缩和总结通过将细粒度情节记录转换为更紧凑和抽象的表示来满足这一需求，减少冗余并提高记忆效率。在实践中，许多 Agent 系统定期或在任务完成后执行总结，将交互历史整合为适合长期存储的语义或分层记忆（Wang等人，2025j；Chen等人，2025d）。

分层巩固进一步将压缩记忆组织为多级或树结构表示，支持跨不同抽象级别的可扩展检索（Sarthi等人，2024；Rezazadeh等人，2025b）。Dynamic Cheatsheet进一步实例化这一思想，通过维护一个持续更新的紧凑的、任务自适应的摘要，仅表面最显著的信息供持续推理使用，减少推理时重复的大规模检索和上下文扩展（Suzgun等人，2025）。

虽然这些机制改善了上下文利用和可扩展性，先前工作强调了抽象保真度与长期回忆之间的固有权衡，使总结策略成为单 Agent 记忆系统中的关键设计考虑（Lee等人，2024；Wu等人，2025d）。

4.1.5 遗忘与保留

在 Agent 系统中，记忆相关性随时间演化，随着任务目标变化和环境变得非平稳，不加区分的记忆积累越来越与有效的长期推理不一致。遗忘机制通过显式移除或抑制不再与 Agent 持续目标一致的记忆条目来减少过时或低效用信息的影响。在实践中，遗忘通常通过启发式策略（如基于新近度的衰减或重要性阈值）以及学习到的策略来实现，这些策略在显式资源或效率约束下优化记忆移除（Alla等人，2025；Wang等人，2025q）。

相比之下，保留机制决定哪些记忆应在扩展交互时程上被保存和优先考虑，确保任务相关知识在持续记忆增长下仍可访问。近期工作强调自适应保留策略，其中 Agent 基于任务上下文、反馈信号或长期性能目标动态调整保存优先级，支持在非平稳环境中实现稳健行为（Yan等人，2025b；Zheng等人，2025a）。

4.2 多 Agent 系统的记忆操作

在多 Agent 系统中，记忆操作机制描述了多个 Agent 如何在协作过程中共同构建和重用记忆：每个 Agent 可以保持自己的私有记忆，它们也可以通过共享工作空间共享经验。与单 Agent 系统一样，此机制仍包括索引、检索、更新、压缩、总结、遗忘和修剪等基本操作。它还可以额外支持多个 Agent 贡献的经验的聚合和蒸馏到共享知识和技能库中，允许一个 Agent 发现的有用策略惠及其他 Agent 或未来协作。

除了这些基本操作，在多 Agent 设置中更重要的是跨 Agent 读写规则：在每个任务中，系统需要为具有不同角色的 Agent 选择合适的记忆。这些规则不仅决定了每个 Agent 可以访问哪些记忆，还决定了角色特定经验、专业技能和执行反馈如何跨 Agent 转移。此外，系统通常需要额外操作来消除冗余、解决冲突并保持记忆一致性。这些操作必须考虑由异构 Agent 生成的记忆的来源、可靠性和兼容性，特别是当它们的观察或学习策略冲突时。

通过选择性共享、验证、巩固和修订，多 Agent 记忆可以在保持专业能力和连贯的长期协作的同时支持集体自我进化。

4.2.1 记忆架构

在多 Agent 系统中，记忆可以以不同方式组织，在存储层、跨 Agent 读写权限定义方式以及系统是否引入控制器方面有所不同。这些设计选择决定了路由是否能构建高效的记忆视图，以及信息泄露等系统性问题是否可能出现。

仅私有。在仅私有架构中，每个 Agent 有其自己的独立记忆，其读写权限仅在该 Agent 的记忆内生效。在多 Agent 协作中，Agent 只能依赖自己的记忆和当前任务上下文来工作。这种设置提供了强隔离性并使系统更易于检查。然而，相同记忆通常会在多个私有空间中重复创建，可能浪费资源。

代表性示例包括RecAgent（Wang等人，2025h），它为每个用户实例化一个 Agent 并保持私有记忆以避免混淆不同用户历史以更好地保护隐私；TradingGPT（Li等人，2023b），其中每个交易 Agent 维护自己的记忆以遵循一致的风险偏好和行业聚焦，协作主要通过交换选定观点而非共享完整记忆进行；以及MetaAgents（Li等人，2025j），它为每个角色配备其过去思想和决策的隔离记忆，使角色保持稳定一致， 而从对话中获得的任何信息都被本地写回。

共享工作空间。与仅私有记忆不同，共享工作空间设计使用所有 Agent 都可以读写的公共池。Agent 通过此共享状态共享中间结果，因此不需要繁重的点对点消息传递，这降低了通信成本。然而，共享池可能迅速变得嘈杂，因此需要过滤机制以及协调策略以避免多个 Agent 同时更新相同信息时发生冲突。

作为代表性共享工作空间设计，MetaGPT（Hong等人，2024）使用简单共享池发布角色 Agent 的中间产物，每个 Agent 应用其角色档案过滤池并仅将相关记忆拉入上下文。IntRecAgent（Huang等人，2025b）通过建立Candidate Bus使共享状态更具任务特定性：工具反复读取当前候选集并写回过滤后的候选，因此集合逐步缩小，可以避免提示长度溢出。

MAICC（Jiang等人，2025d）进一步将工作空间扩展为具有离线数据和在线重放缓冲区的共享经验池，Agent 使用当前子轨迹查询并检索前k个相似轨迹以在上下文中重用。

混合。此设置同时保持私有层和共享层。它使用策略决定新信息是写入私有空间还是共享空间。在每次调用时，它为 Agent 构建一个权限受限的记忆视图，因此 Agent 只看到其被允许看到的内容。这提供了最大化记忆重用与保持敏感信息隔离之间的平衡。例如，在 Collaborative Memory（Rezazadeh等人，2025a）中，系统将所有记忆片段分离为私有记忆和共享记忆。

在写入期间，写入策略决定信息是普遍有用还是用户特定，然后将其写入共享或私有层。在读取期间，读取策略使用系统提供的访问图为当前调用构建权限受限的记忆视图。类似地，在MirrorMind（Zeng等人，2025）中，每个Author Agent 维护对应其研究兴趣的私有记忆，同时通过公共记忆共享基础学科知识。这种混合设计形成了AI Scientist的结构。

编排式。前三个类别主要在记忆存放位置和谁可以看到方面有所不同。相比之下，编排式设计引入了一个显式控制器，在分层工作流中协调 Agent 并调节记忆访问。具体地，控制器分解任务，将子任务分配给角色 Agent，并决定每个 Agent 可以读写什么。这种集中协调非常适合多阶段工作流和强约束设置，但可能引入瓶颈和额外的系统复杂性。

ChatDev（Qian等人，2024a）通过在预定义ChatChain（设计、编码、测试）下运行角色 Agent 来例证此模式，其中阶段输出作为结构化交接以减少跨阶段上下文过载。MIRIX（Wang & Chen，2025）类似地编排记忆维护：元记忆管理器将更新/检索路由到专门的记忆管理器，并将其结果聚合为统一响应。值得注意的是，此控制层正交于存储布局，可以与仅私有、共享工作空间或混合记忆存储结合。

例如，MGA（Cheng等人，2025）将GUI交互组织为观察、规划和接地 Agent 流水线。规划器充当控制器，而较低级 Agent 可以在共享工作空间中提交中间状态，规划器选择在每个步骤检索和注入什么历史。类似地，AgentFlow也将编排与共享工作空间结合：规划器控制所有模块并将关键中间结果写入共享记忆以供后续检索（Li等人，2025l）。

4.2.2 记忆路由

给定架构，路由描述了一组记忆分配和访问规则。当新任务到达时，系统需要为具有不同角色的 Agent 构建单独的记忆视图：它决定应检索哪些过往记忆以及如何将它们注入每个 Agent 的上下文。我们按路由决策所在位置对路由方法进行分组：集中编排器、个体 Agent 或记忆存储本身通过检索和匹配。

编排器基路由。这指集中编排器以统一方式做出路由决策的设置。它维护全局任务状态和协作进度，将复杂目标分解为子任务，然后基于角色和能力将子任务分配给不同工作 Agent，同时分发所需记忆并决定执行顺序。这些决策可以随任务状态变化动态更新。此方法强调集中全局工作流，但代价是编排器可能成为性能和鲁棒性的瓶颈，产生单点负载和故障风险。

例如，在LEGOMem（Han等人，2025）中，编排器保持全局状态、生成下一个子任务、选择 Agent 执行它，并使用 Agent 的摘要更新状态；记忆以相同方式调度，任务记忆被注入编排器以供规划，子任务记忆被路由到所选 Agent 以供执行。GameGPT（Chen等人，2023）将焦点转向工作流路由：管理器定义多阶段流水线，并要求每个阶段将关键中间输出写入共享工作空间，以便后续阶段可以继承和重用它们。

最后，Westhäußer等人（2025）将编排扩展到记忆源路由，其中使用MCP的编排器选择调用哪些记忆源并注入最相关的片段；如果注入的证据不足，Self-Validator要求更多检索并更新上下文，实现集中记忆控制。

Agent 发起路由。与编排器基路由相比，在此方法中路由决策不是由集中编排器分配。相反，每个 Agent 基于其角色和任务在本地发起它们。信息通常首先发布到共享记忆池中，然后 Agent 使用约束机制选择它们需要的记忆，形成自己的记忆视图。此方法通常更灵活，但也更依赖于良好的过滤设计以避免共享池中的噪声、冲突或遗漏重要信息。

在SRMT（Sagirova等人，2025）中，每个 Agent 保持个人记忆向量并在本地决定从共享记忆中读取多少：在每一步中，它使用对其记忆的交叉注意力从共享记忆序列中选择并更新其个人记忆。采用更显式过滤路径的S^3（Gao等人，2023）将共享记忆视为平台范围的消息流，并用遗忘、相关性和源可信度等因素对项目评分，仅保留一小部分作为 Agent 自身的记忆视图。

在Talker-Reasoner设置（Christakopoulou等人，2024）中，共享存储（mem）由Reasoner写入并由Talker读取，Talker可以选择读取什么或在回复前等待新更新，表明 Agent 可以决定何时从记忆中读取。

记忆驱动路由。与上述不同，路由在此主要通过从记忆存储中检索完成。系统将当前任务表示为查询，然后在记忆存储中执行“检索、评分和重排序、选择”以获得最相关的记忆子集并将其注入上下文。有时它还可以使用记忆之间的结构化链接（如图基扩展）将检索结果扩展为更完整的经验片段集合。

作为记忆驱动路由的典型形式，G-Memory（Zhang等人，2025c）将多 Agent 历史组织为分层图，检索新任务的相关节点，通过邻域链接扩展它们，并将结果压缩为核心子图，然后修剪为角色特定的记忆视图。CRMWeaver（Lai等人，2025）改为在可重用指南级别路由：它从过往成功轨迹中检索最相关的工作流指南并将其注入当前上下文，并在无匹配时写回新指南。

最后，在Spark（Tablan等人，2025）中，每个编码问题被视为查询，检索 Agent 分析意图并从共享存储中选择最相关的文档和过往经验轨迹。

4.2.3 记忆隔离与冲突

基于多 Agent 架构和记忆路由，记忆隔离非常重要，因为记忆由多个 Agent 读写而不是在单个顺序循环中更新。多个 Agent 可能并行产生结论，其可访问资源和权限通常随时间变化。如果系统将所有信息写入同一个共享池而不进行任何分离并使其对每个 Agent 可见，可能会出现一致性冲突：不同 Agent 可能写入相互矛盾的事实，过时信息可能未被移除并仍被检索，可能误导推理。

Agents Thinking Fast and Slow（Christakopoulou等人，2024）报告Talker可能产生错误或仓促答案，因为它可以在Reasoner更新之前读取过时的信念状态。这里的记忆冲突主要通过两种方式处理：在写入时控制，或通过迭代循环逐步改善一致性。

用于记忆隔离的写控制。一种直接策略是在记忆写入和更新阶段强制隔离。在每个交互轮次中，Agent 首先将新提取的候选事实与现有记忆状态进行比较，并通过受控评估机制选择性更新记忆，而非盲目追加新信息。在Memory-R1（Yan等人，2025b）中，记忆管理器 Agent 是唯一被允许修改记忆的 Agent。这包括四种离散编辑动作，即ADD、UPDATE、DELETE和NOOP。

ADD写入新条目；UPDATE重写或合并现有条目（特别是当新信息是对旧信息的精炼或修正时，系统倾向于保留信息更多的版本）；DELETE移除明显被新证据反驳或过时的条目；NOOP表示信息已被覆盖或不重要，因此不进行更新。对于每个用户查询，记忆管理器观察当前记忆状态并决定操作哪些特定记忆槽。结果，无关记忆条目保持不变。

WebCoach（Liu等人，2025b）采用了写时隔离的相关形式，其中记忆存储仅在事件完成后更新，因此部分轨迹永远不会被写入长期记忆，这隔离了事件并防止了记忆冲突。

带反馈循环的记忆一致性。与写控制相比，这条工作线将记忆冲突视为一致性迭代优化过程。通常，多 Agent 记忆系统在稳定的约束记忆中强制执行任务的硬性要求，后续迭代可以查阅，并保持增长的反馈记忆，记录早期轮次的失败以支持后续迭代中的学习。EvoMem（Fan 等人，2025b）是一个代表性示例。其验证器将每个候选与存储的约束记忆进行比较并输出分数。系统仅在分数达到一百时接受解决方案。

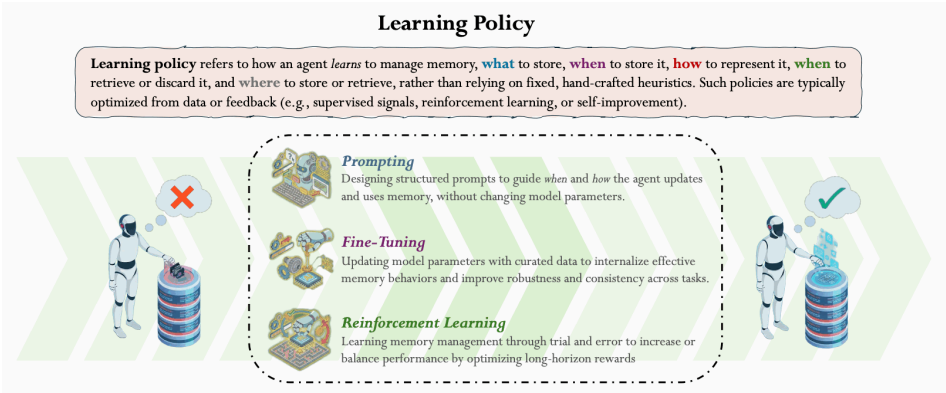

图7：基础模型 Agent 记忆的进化与优化策略。提示、微调和强化学习分别从外部控制、参数内化和长期反馈三个层面改进记忆决策。

5 记忆进化与优化策略

本节考察 Agent 如何通过包括读取、写入、更新、巩固、抽象、遗忘和修剪在内的操作来获取进化和优化记忆的策略。这些策略不是依赖硬编码的启发式，而是决定了何时以及记住什么、积累的经验应如何转换为结构化知识和可重用技能，以及现有记忆应如何根据后续交互和反馈进行修订。我们基于其主要优化机制将这些策略分类为三种范式：提示、微调和强化学习，如图7所示。

这些范式在不同层面启用记忆进化和优化。提示直接重组织和精炼推理时的外部记忆；微调将记忆操作策略内化为模型参数；强化学习根据其对 Agent 执行和任务结果的长期影响来优化记忆决策。

5.1 提示驱动的记忆进化与优化

提示驱动的记忆优化指使用提示、反思或总结来管理记忆而不修改模型参数的推理时策略，区别于微调（§ 5.2）和强化学习（§ 5.3）。此范式将记忆策略参数化为自然语言提示。Agent 执行这些提示以确定何时访问、修改或修剪记忆。此方法的主要优点包括消除昂贵的模型微调和高策略可解释性。我们进一步将此范式分类为静态提示控制（§ 5.1.1）和动态提示控制（§ 5.1.2）。

5.1.1 静态提示控制

静态提示控制将记忆策略编码为在执行期间保持不变的固定的人工设计规则。记忆决策在设计时通过提示模板或预定义模式指定，提供强可解释性和可预测行为，但缺乏基于交互反馈或分布漂移进行适应的能力。

现有静态提示记忆控制可按三个设计目标分组：

- **静态记忆操作系统与组织。** 记忆被视为固定形式的结构化容器，通过分层分区、索引、摘要或模式化表示来减轻长时程上下文退化。代表性工作包括 SCM、MemGPT、LiCoMemory、MemoChat、A-Mem、D-SMART、MemWeaver 和 Zep。
- **单 Agent 的静态记忆控制。** 访问与保留由提示中的角色身份或领域先验约束，而不是由学习得到的相关性信号决定。代表性工作包括 RoleLLM、ChatHaruhi、Mem-PAL、WarAgent、MemoTime、FinMem、TradingGPT 和 MemoCRS。
- **多 Agent 的静态记忆分配与协调。** 记忆通过预定义角色、模块化分解和结构化通信协议分布在不同 Agent 之间，无需学习式协调。代表性工作包括 MIRIX、LEGOMem、G-Memory、MADRA、GameGPT 和 ChatDev。

5.1.2 动态提示控制

动态提示控制探讨编码在自然语言提示中的记忆策略是否可以在测试时基于经验和反馈进行适应，而不更新模型参数。此范式不将记忆行为固定在设计时，而是将记忆控制视为语言中介且持续可修订的过程。

此领域的现有工作围绕一组密切相关的研究问题。一个研究方向询问记忆使用策略是否可以通过对过往结果的反思来纠正，提示 Agent 分析失败或成功并将这些见解转换为修订的记忆指令以指导未来行为（例如，Reflexion（Shinn等人，2023）、ReasoningBank（Ouyang等人，2025）、WebCoach（Liu等人，2025b）和QuantAgent（Wang等人，2024g））。

另一个研究方向调查记忆表示本身是否可以动态优化以在有限上下文预算下提高信息效率，将压缩、去噪和结构重组视为自适应的、提示驱动的过程（例如，ACON（Kang等人，2025c）、ACE（Zhang等人，2025j）、SeCom（Pan等人，2025）、Nemori（Nan等人，2025）、CAM（Li等人，2025d）、EvoMem（Fan等人，2025b）和ViLoMem（Bo等人，2025））。

进一步的问题涉及动态提示是否可以将积累的经验蒸馏为可重用的程序性知识，如推理模板、执行脚本或工具使用策略，这些超越了情景回忆（例如，BoT（Yang等人，2024b）、Memp（Fang等人，2025b）和ToolMem（Xiao等人，2025））。尽管具有自适应性，这些方法从根本上仍是语言中介的，缺乏显式信用分配，与微调和基于强化学习的方法相比，限制了其长期策略优化的能力。

5.2 参数化记忆策略的微调

超越基于提示的适应，监督微调将记忆策略内化为模型参数，支持更稳定和可重用的记忆行为。从策略学习视角来看，基于SFT的方法研究记忆策略一旦嵌入模型权重后如何被内化、稳定和高效执行。

5.2.1 策略内化为参数

SFT基记忆控制的一个决定性特征是记忆策略被内化为模型参数，将记忆从外部上下文操作问题转变为参数化策略表示。这些方法不依赖推理时的提示或显式缓冲区，而是将记忆相关行为直接嵌入权重空间，支持跨任务的稳定和可重用记忆使用。

在此范式中，现有工作探索了几个密切相关的研究问题，涉及记忆控制策略如何嵌入模型参数以及这种内化应如何结构化。一些方法关注内化记忆内容本身，将短期上下文信息蒸馏为长期参数化表示（Wang等人，2024b；2025p）。

其他方法强调通过学习参数化接口或轻量级模块来内化记忆访问和检索行为，这些模块调节与外部或结构化记忆存储的交互，而不是将原始内容吸收到权重中（例如，Memory3（Yang等人，2024a）和MLP Memory（Wei等人，2025c））。一条相关工作线研究这种参数化记忆策略如何分层组织，支持大规模记忆组件的可扩展调用，同时将长尾知识与核心推理能力解耦（Pouransari等人，2025）。

尽管存在差异，这些方法共享将记忆控制从提示级操作转移到通过监督学习的参数化决策规则的目标。

5.2.2 参数化策略稳定与边界控制

除了内化，SFT通过学习一旦记忆控制嵌入参数后对应该写入、纠正或抑制什么的显式边界来实现记忆策略的稳定。这些方法不仅旨在扩展记忆容量，还旨在防止长期使用下的错误积累、概念漂移和角色不一致。

这些工作的一个共同主题是使用监督来正则化记忆更新并强制执行边界约束。一些方法训练模型在将信息提交到记忆之前执行反思或自我分析，鼓励存储与任务意图对齐的高层、抗噪声表示（Liu等人，2023b；Zhang等人，2025r；Chen等人，2025d）。

其他方法强调通过学习何时应通过纠正信号、验证器反馈或路由机制修订或覆盖现有知识来识别和修复错误或过时记忆（例如，WISE（Wang等人，2024f）、SuperIntelliAgent（Lin等人，2025）和CRMWeaver（Lai等人，2025））。在长时程交互设置中，防御性边界控制通过限制哪些经验可以被保留或重用来进一步约束记忆更新以保持角色或身份一致性（例如，Character-LLM（Shao等人，2023））。

共同地，这些方法将反思和自我纠正视为通过监督学习稳定和有界记忆策略的机制，而非孤立的提示级技术。

5.2.3 参数化策略效率与检索优化

除了学习存储什么以及如何稳定记忆，SFT还用于精炼参数化记忆策略在推理时的执行方式，特别是在记忆读取和检索期间。这些方法不依赖穷举上下文访问，而是将检索本身视为学习策略，并优化查询如何被制定、迭代精炼并应用于压缩记忆表示。

通过监督训练，模型学习为针对性记忆访问生成精确检索线索（Qian等人，2025b），执行多跳或渐进式检索以跨推理步骤精炼查询（Ko等人，2024），并优化压缩感知检索通过内化或可逆精炼记忆表示（Cao等人，2025b；Wang等人，2025m），从而在最小化计算和存储开销的同时提高推理鲁棒性。

5.3 记忆策略的强化学习

与提示设计和SFT不同，提示设计在推理时操作外部记忆而无需参数更新，SFT通过监督将记忆策略内化为权重，强化学习将记忆优化重新框定为顺序决策问题，其中 Agent 学习长期导向的记忆策略，通过试错与其推理和行动循环整合。与SFT基方法相比，强化学习具有两个关键优势：它直接优化不可微分的记忆效用指标（如检索精度、上下文效率和下游任务成功率），并学习适应任务上下文和动态的隐式策略，而非复制静态人工标注行为。

我们将基于RL的记忆策略研究按优化范围分组：单轮上下文优化、跨事件和多 Agent 记忆策略。在第一层，强化学习用于优化给定固定交互事件或会话内的即时记忆使用。在这些设置中，Agent 必须决定在单个任务或对话中保留什么上下文以及如何管理其工作记忆。

这包括用于KV缓存管理的强化学习策略，Agent 学习在解码期间驱逐什么令牌以平衡速度和保真度，以及用于选择和压缩外部记忆的策略以最大化给定上下文预算内的下游任务性能（例如，CURATOR（Zhang等人，2025p）、MemAgent（Yu等人，2025b）、Tova（Jeunen等人，2025）和MEM1（Zhou等人，2025c））。

在第二层，强化学习优化跨多个事件或会话持续的记忆策略。当 Agent 必须跨多个任务保留、巩固和重用经验时，强化学习变得特别有价值，因为长期和跨事件奖励信号提供了关于哪些记忆应由记忆策略持久化、适应或修订的显式信用分配。此层面的研究关注一旦记忆跨多个事件或 Agent 后，经验应如何被表示、重用和协调。

这些方法不是保留原始交互历史，而是旨在蒸馏更高级别的决策相关知识，如可重用策略、自我纠正规则或抽象行为模式，其效用通过重复强化信号进行评估。此视角构成了MCTR（Li等人，2025h）和基于图的经验抽象（Xia等人，2025）等方法的基础，这些方法将经验编码为通过交互学习的可转移决策知识。

至关重要的是，这种经验以由强化学习支配的上下文相关方式检索和应用，如Retroformer（Yao等人，2024）和Memento（Zhou等人，2025a）中的反思性检索策略所例证。随着记忆进一步跨 Agent 或表示空间扩展，强化学习使反馈能够传播到单个轨迹之外。

这包括潜在或非文本记忆表示如MemGen（Zhang等人，2025d）、ReMemR1（Shi等人，2025a）中的检索路径优化和回调机制，以及多 Agent 系统中的共享或去中心化记忆策略如MAICC（Jiang等人，2025d）和SRMT（Sagirova等人，2025）。这些研究将跨事件和多 Agent 记忆框架为基于强化学习的记忆控制的最广泛范围，其中记忆策略通过积累的交互和奖励反馈而非预定义规则或一次性监督进化。

6 进化与扩展：记忆、上下文与环境

随着 LLM 和 Agent 部署在日益现实的环境中，其操作期间积累的经验和上下文信息沿开放世界环境中的三个扩展轴快速增长：交互时程、环境复杂度以及交互环境、工具和 Agent 的数量。虽然传统评估通常依赖静态、上下文受限的设置，忽视了环境动态，但现实世界效用要求 Agent 跨扩展时间框架和异构数据结构积累、保留、巩固和更新知识。

简单地扩展上下文窗口是不够的，因为长时间运行交互会引入冗余、过时、冲突和日益异构的经验，这些必须被选择性组织和转换。本节探讨记忆如何作为应对这一扩展挑战的基本架构解决方案出现，从简单交互日志转变为管理增长经验和支持持续 Agent 进化的自适应载体。通过选择性保留任务相关状态、将轨迹抽象为可重用知识和技能、跟踪演化用户偏好以及跨 Agent 协调经验，可扩展记忆支持在开放世界环境中实现长时程执行、持久个性化和集体适应。

6.1 上下文受限的简单环境

今天大多数 LLM 和 Agent 基准仍然配置在上下文受限的简单环境中，Agent 被置于紧凑封闭的世界中，仅在短时程任务实例上交互（Hendrycks等人，2021b）或与合成的、非真实用户交互（Maharana等人，2024）。虽然此类设置有利于可重复性和实验比较，但它们严重低估了现实世界 Agent 面临的记忆需求。

因此，强基准性能通常反映了上下文内模式匹配或短期推理的熟练程度，而非跨扩展交互和演化用户偏好与背景积累、保留和重用知识的能力。这种不匹配导致了显著的实用性差距：在现有基准上取得高分的 Agent 经常未能在开放、动态环境中展示长期适应、用户个性化或任务特定技能重用。

现有 LLM 和 Agent 评估的很大一部分仍然局限于静态、上下文受限的环境。

经典的通用问答基准，包括SQuAD（Rajpurkar等人，2018）、HotpotQA（Yang等人，2018）和KILT（Petroni等人，2021），在冻结的维基百科快照上运作，而MS MARCO（Craswell等人，2021）、Natural Questions（Kwiatkowski等人，2019）、SearchQA（Dunn等人，2017）和TriviaQA（Joshi等人，2017）用预收集的查询和段落取代实时信息访问。

这些设计产生稳定、有界且良好受控的信息源，现代系统在此达到近乎饱和的性能。然而，检索和推理严格是情景的且实例隔离的：Agent 既不维持跨查询状态，也不面对时间漂移、源不一致或演化知识。因此，此类基准对持久记忆施加了最小要求，主要测试短期检索和上下文内推理而非长期知识积累或自适应上下文管理。类似的限制延伸到工具使用和Web交互基准。

诸如ToolBench（Qin等人，2024）、WebShop（Yao等人，2022）和WorkArena（Drouin等人，2024）等框架依赖固定API或自托管环境，抽象掉认证、故障恢复和长期用户或任务状态。即使是与实时Web交互的系统，包括WebGPT（Nakano等人，2021）和WebVoyager（He等人，2024a），仍受限于受限浏览界面、筛选与组织网站列表和严格交互预算。

ScienceWorld、ALFWorld、TextWorld 和 Jericho 等顺序交互环境，以及 APPS、SWE-bench、SWE-Bench Pro 和 SWE-Lancer 等编程基准，提高了任务复杂度，但仍以单次情景、任务重置和短期评估为中心。

因此，这些基准仍难以评估跨事件的知识积累和长时程一致性，也难以揭示记忆对稳健现实世界 Agent 部署的关键作用。

6.2 上下文规模爆炸的现实世界环境

与研究环境中的上下文受限基准评估不同，现实世界部署将 Agent 暴露于上下文沿多个轴扩展的环境。它随长时程交互积累，由于结构化和异构环境状态而变得越来越复杂，并跨越多个环境、Agent 和工具。此外，现实世界上下文不仅数量增加：它持续变化，因为用户修订其偏好、环境演化、工具产生新观察以及 Agent 获取新知识和技能。可扩展记忆因此必须同时支持增长上下文的压缩和 Agent 保留状态的持续进化。

6.2.1 随长时程交互进化

跨扩展交互时程的用户记忆进化。在持久用户-Agent 交互中，Agent 需要维持与用户跨扩展对话时程对齐的连贯行为和决策一致性。与短的自包含对话不同，早期轮次引入的信息，如用户偏好、身份属性或隐式任务约束，可能不会立即影响响应，但通常在交互后期阶段变得决定性。因此，交互历史更多地作为用户状态的演化表示，而非静态输入，必须随时间选择性维护和修订。

这种信息持久性需求源于某些用户属性相对稳定以及偏好和目标跨不同轮次的逐步演化（Li等人，2016）、涉及规划或持续目标的对话任务的长期结构，以及身份和目标一致性对用户信任和可用性的重要性（Zhang等人，2018）。

长期个性化因此必须平衡一致性和适应：Agent 应保留持久用户特征，同时识别近期交互何时提供足够证据来更新先前推断的偏好或约束。然而，在固定上下文窗口约束下，长时程对话系统经常表现出如早期上下文遗忘和随交互长度增加而渐进上下文漂移等故障模式（Liu等人，2024b；Xu等人，2022a）。这些故障不是孤立的推理错误，而是长期上下文管理不善的累积后果。它们可能导致 Agent 要么忘记持久偏好，要么锚定于过时信息，两者都削弱了持续偏好对齐。

现有的缓解策略，如滑动上下文窗口、启发式截断和基于摘要的记忆机制（Park等人，2023），以及显式长期结构如用户档案或角色记忆（Xu等人，2022a），改善了可扩展性但往往降低回忆可靠性。在实践中，看似外围的信息可能被不可逆地丢弃，尽管其具有未来潜在相关性，暴露了上下文预算控制与长期信息可访问性之间的基本权衡（Liu等人，2024b）。因此，挑战不仅是保留更多交互历史，而是持续确定哪些信息保持稳定、哪些已过时、以及哪些应修改 Agent 当前对用户的表示。

多轮工具使用的积累执行记忆。上下文规模爆炸在依赖多轮工具使用和推理架构的 Agent 中可能加剧。除了对话历史，基于工具的 Agent 必须保留工具输入、执行输出和中间状态，其中许多在后续推理步骤中被重复引用（Schick等人，2023）。

在ReAct（Yao等人，2023）和Planner-Executor架构（Wang等人，2023b）等框架中，这种积累尤为严重：规划轨迹、工具反馈和反思推理被显式保留以维持连贯性和决策一致性（Shinn等人，2023）。因此，上下文大小可随交互长度和工具调用的数量或复杂度快速增长。

在ReAct风格 Agent 中，显式推理轨迹本身成为上下文的一部分，支持可解释性和复杂推理，但同时引入大量且持久的上下文负担。此外，对于许多现实世界应用，大型或异构工具输出，如Web搜索结果（Wei等人，2025e）和数据库查询结果（Jing等人，2025），也可能急剧积累。天真地移除或压缩这些轨迹可能破坏推理和动作步骤之间的因果依赖关系，并损害后续决策（Yao等人，2023）。

有效的记忆必须转而保留任务关键状态、未解决依赖关系、先前工具效果和执行失败，同时压缩冗余观察和推理轨迹。

因此，虽然多轮工具使用显著增强了 Agent 能力，它也暴露了有限上下文窗口上的可扩展性限制，突显了长时程 Agent 设计的核心挑战（Liu等人，2024b）。除了维护正在进行的任务状态，Agent 还应将重复的工具使用经验巩固为可重用的过程和技能，使未来任务能够受益于先前执行，而非重新产生相同的长推理轨迹。

6.2.2 随环境复杂度进化

随着环境复杂度扩展，Agent 的上下文必须容纳异构数据模态、异步工具交互以及协议或权限约束的外部状态。在此类设置中，上下文不能再被视为附加到提示的线性交互轨迹。现实世界部署需要推理结构化产物，如API响应、文件、数据库、日志和配置状态，其语义取决于模式、来源、更新规则和时间有效性。直接将这些产物展平为令牌序列既低效又在结构上有损，削弱了可解释性、精确检索和定向更新（Modarressi等人，2024）。

因此，增加的环境复杂度将记忆从提示积累的隐式副产品转变为负责结构化上下文管理的显式系统级组件。记忆挑战因此从仅仅保留过往信息转变为跨多样化来源和生命周期维护连贯的、可查询的和可更新的环境状态表示。对于自我进化 Agent，这些表示必须进一步适应模式、接口、权限和环境动态的变化，同时保留先前获得的知识和技能的有效性和来源。

最近的 Agent 系统通过将记忆外化到提示之外并暴露将存储与推理解耦的显式读写接口来解决此挑战（cauri，2025）。外化记忆支持模式感知检索、版本控制、定向编辑和访问控制——这些操作在仅基于提示的上下文中难以或不可行（Yakobi & Sadon，2025）。

基于协议的接口如Model Context Protocol（Anthropic，2024）标准化了 Agent 如何访问外部工具和上下文资源，而技能导向的抽象如Claude Agent Skills（Anthropic，2025）将程序性指令、可执行资源和领域特定工作流打包为可重用单元。共同地，这些抽象将持久状态、环境访问和可重用技能与瞬时推理上下文分离，从而改善模块化、可审计性和行为安全性（Keen，2025）。

随着环境复杂度增加，持久性变得不可避免：用户模型、系统配置和任务产物代表必须跨会话保持一致的持久状态，同时尊重隐私、权限和回滚等治理约束（Sarin等人，2025；Wu & Shu，2025）。这些需求引入了额外挑战，包括模式漂移、并发更新和路径依赖评估，突显了对将记忆视为复杂开放世界环境中上下文管理的核心基础设施的原则性抽象的需求。

因此，记忆必须在内容和结构两个层面进化：它应更新环境事实和任务状态，在模式变化时重组表示，并在其底层接口或动作前提条件不再有效时修订先前学到的技能。

6.2.3 跨多个环境和 Agent 进化

记忆作为跨工具环境的接口。在开放世界设置中，上下文复杂性不仅源于在单个环境内的延长交互，还源于需要跨多个异构工具环境运作，每个由不同的状态表示、动作空间和访问协议定义。例如，个人具身助手可能在物理家庭环境（执行以感知反馈为依据的长时程行为）和数字Web环境（通过基于浏览器的交互检索信息如天气预报）之间交替。

支持这种行为需要能够跨多样动作和观察模态保留、区分和协调环境特定状态的记忆机制（Glocker等人，2025；Hong等人，2025）。在OSWorld（Xie等人，2024）中，GUI Agent 可能在浏览器中发出搜索查询、从新闻应用检索结果、并在文件查看器中查阅文档，每次交互产生环境特定的观察和状态转换。虽然工具支持每个环境内的局部交互，记忆提供了跨环境边界持久化、组织和上下文化信息的接口功能。

因此，记忆系统不仅需要记录过往工具调用，还需要维护环境状态的结构化和分离表示，以支持长时程推理而不超出上下文窗口（Burtsev等人，2020；Rae等人，2019）。

多用户与多 Agent 系统中的记忆。随着 Agent 系统扩展到包括多个 Agent 和人类参与者，有效协调日益依赖结构化通信和共享记忆机制，而非天真的上下文共享，后者在有限上下文窗口下迅速碎片化（Chen等人，2025f）。近期方法引入了 Agent 对齐或半共享记忆抽象，编码关系历史、跨 Agent 依赖和跨长时程交互的任务相关状态。

例如，Intrinsic Memory Agents为 Agent 配备角色特定记忆模板，保留专业视角同时支持整合到共享上下文载体中，显著改善长时程规划稳定性（Yuen等人，2025）。除了记忆，结构化通信协议在协调中发挥核心作用：分层和角色感知对话方案减少跨 Agent 交换中的噪声和偏差（Wang等人，2025s），而认知自适应编排框架基于推断的协作者状态动态调整通信模式（Zhang等人，2025h）。

在涉及多个人类和 Agent 的开放世界部署中，协调进一步扩展到分歧下的对齐、冲突解决和集体决策。

实证研究表明，基于辩论的协议和决策规则如共识或多数投票显著影响跨推理和知识任务的性能（Kaesberg等人，2025；Samanta等人，2025），包括多模态提取设置，其中少数辩论轮次可衡量地提高准确性（Huang & Caragea，2026），而自适应和无共识辩论机制平衡计算成本、鲁棒性和顺从效应（Fan等人，2025a；Cui等人，2025b）。

随着这些系统扩展，显式图结构表示日益支撑编排和通信，将关系依赖组织为可优化的通信拓扑，并将上下文管理重新框定为开放世界系统中的分布式记忆、协调和对齐问题（Qian等人，2025a；Zhang等人，2025f；e）。

7 评估

评估基础模型 Agent 记忆从根本上关于衡量存储的信息和经验在长时程交互下是否准确、有用、可靠且高效可访问。下面，我们在第7.1节中总结三种不同基本类别中常用的指标，包括基于准确率的、基于相似度的和 LLM 作为评判者。此外，我们在第7.2节中收集常用的基准来评估基础模型 Agent 的性能。

表3：基础模型 Agent 记忆评估中使用的指标。

指标	简要描述	代表性基准
基于准确率的指标		
准确率/记忆准确率	在基准粒度（如问题、轮次、会话或任务级）计算的正确回答实例的比例。	HotpotQA（Yang等人，2018）、PerlTQA（Du等人，2024）、MemoryBank（Zhong等人，2024）
F1分数	精确率和召回率的调和平均，在令牌级计算并可选择聚合到问题、轮次或会话级。	MuSiQue（Trivedi等人，2022）、MSC（Xu等人，2022a）、PerlTQA（Du等人，2024）
Recall@K	通过检查相关证据是否出现在前 K个结果中来评估检索成功。	LongMemEval（Wu等人，2025b）、PerlTQA（Du等人，2024）
平均精确率均值（MAP）	对相关项排名位置的精确率进行平均，然后跨查询平均。	PerLTQA（Du等人，2024）、PMR（Kohar & Krishnan，2025）
NDCG@K	通过优先考虑顶部位置来衡量截止K处的排名相关性。	LongMemEval（Wu等人，2025b）、PMR（Kohar & Krishnan，2025）
成功率（SR）/目标完成（GC）	在环境定义的成功检查器下完成的交互任务的比例，通常在任务或事件级测量。	WebArena（Zhou等人，2024）、OSWorld（Xie等人，2024）、MineDojo（Fan等人，2022）
Pass@K / 解决率（RR）	K次采样尝试中至少一个正确解决方案出现的概率（Pass@K），或端到端解决的问题（RR）。	HumanEval（Chen等人，2021）、SWE-Bench（Jimenez等人，2024）
记忆完整性（MI）	记忆提取的完整性，通过记忆点上的覆盖率或召回率衡量。	HaluMem（Chen等人，2025a）
虚假记忆率（FMR）	引入幻觉记忆的比率，包括捏造或不正确的更新。	HaluMem（Chen等人，2025a）
基于相似度的指标		
ROUGE	测量生成文本与参考文本之间的n-gram和最长公共子序列重叠。广泛用于摘要。	LoCoMo（Maharana等人，2024）、MemoryBench（Ai等人，2025）
BLEU	与参考的n-gram精确率重叠。常用于对话生成。	DuLeMon（Xu等人，2022b）、LoCoMo（Maharana等人，2024）
Distinct-n	计算唯一n-gram的比率以测量词汇多样性和话语重复。	DuLeMon（Xu等人，2022b）、MADial-Bench（He等人，2025a）
BERTScore	候选与参考之间的基于嵌入的语义相似度。	LoCoMo（Maharana等人，2024）、MADial-Bench（He等人，2025a）
FactScore	事实级忠实度：提取原子声明并测量受检索证据支持的比例。	LoCoMo（Maharana等人，2024）、Face4Rag（Xu等人，2024b）
困惑度	预测参考文本的基于似然的指标，通常在令牌级并跨会话聚合。	MSC（Xu等人，2022a）、DuLeMon（Xu等人，2022b）
LLM 作为评判者指标		
响应正确性	强 LLM 判断响应是否回答查询或满足约束。	LongMemEval（Wu等人，2025b）、MemTrack（Deshpande等人，2025）
忠实度/证据支撑度	评判者检查响应声明是否得到检索上下文或记忆支撑，或检查引用是否支持声明。	SeekBench（Shao等人，2025）、LiveResearchBench（Wang等人，2025f）
偏好遵循	评判者通过检查输出是否满足用户陈述的偏好或约束来评估偏好遵循。	PrefEval（Zhao等人，2025c）、BEAM（Tavakoli等人，2025）、Convomem（Pakhomov等人，2025）

7.1 指标

表3总结了基础模型 Agent 记忆评估中最常用的指标。评估基础模型 Agent 及其记忆模块性能有多个维度。有些任务有清晰的真值答案，可以用精确正确性评分（Du等人，2024；Dunn等人，2017），而其他任务是开放式的（对话（Budzianowski等人，2018）、摘要（Maharana等人，2024）、偏好遵循（Zhao等人，2025c）），其中多个输出可接受且基于参考的评分不可靠。

因此，现有工作通常结合结果级正确性与检索导向评估和基于评判者的评分标准，取决于基准是否暴露记忆模块、使用长上下文提示，或评估在特定环境、场景或任务中行动的 Agent（Patlan等人，2025；Yadav等人，2025）。

基于准确率的指标。当任务有清晰的目标结果时，使用基于准确率的指标。对于关于长历史的问题回答，准确率或记忆准确率直接评估最终响应与真值答案的对齐（Yang等人，2018；Zhong等人，2024；Du等人，2024）。F1通过给予令牌级部分重叠信用来放宽精确匹配标准。当响应不完全匹配但有一定差异时尤其普遍（Trivedi等人，2022；Deng等人，2023）。

当基准评估显式记忆模块时，Recall@K、平均精确率均值（MAP）和归一化折损累计增益（NDCG）@K变得核心，因为它们区分"系统是否检索到正确证据"与"生成器是否很好地表述了答案"。Recall@K衡量相关项出现在前 K个检索结果中的比例（Wu等人，2025b）。MAP和NDCG@K同时考虑排序质量，并给予首先放置相关记忆的系统更高分数（Kohar & Krishnan，2025）。

对于交互 Agent，正确性由环境评估器定义，因此成功率（SR）或目标完成（GC）代表 Agent 是否端到端完成任务（Zhou等人，2024；Zheng等人，2025a）。对于具身 Agent，路径长度加权成功率（SPL）、导航效率和交互成功率等额外指标常用于捕捉超越二元完成的执行质量。

对于代码和工具使用设置，Pass@K和解决率（RR）衡量K次采样尝试中是否至少有一个解决任务或问题（Yao等人，2025；Jimenez等人，2024）。此外，记忆中心基准越来越多地添加难以仅用端任务准确率捕捉的故障模式评估。记忆完整性量化提取的记忆是否覆盖所需记忆点，虚假记忆率衡量系统在存储、更新或使用过程中引入捏造或不正确记忆的频率（Chen等人，2025a）。

基于相似度的指标。基于相似度的指标在对话生成和摘要中最普遍，其中输出是自由形式的，正确性不能完全通过准确率或F1分数检查（Chen等人，2024d）。BLEU、ROUGE和ROUGE-L测量与参考的词汇重叠，有助于跟踪表面相似性，但可能低估有效释义并高估流畅但缺乏依据的响应（Gehrmann等人，2023；Ai等人，2025）。

Distinct-n通过测量词汇多样性来补充重叠指标，阻止可能夸大相似度分数而不改善忠实度的重复生成（He等人，2025a）。当词汇重叠过于严格时，BERTScore提供基于嵌入的语义相似度近似（Maharana等人，2024），FactScore通过检查原子事实声明层面的一致性来评估记忆忠实度（Min等人，2023），这在摘要用作长历史压缩机制时尤其相关（Saxena等人，2025）。

困惑度也被一些长对话基准用作跨会话生成质量的基于似然的指标（Xu等人，2022a），但它仍然是记忆性能的间接指标，因为它不验证模型内容是否由正确的历史证据支撑（Durmus等人，2022）。

LLM 作为评判者指标。当真值答案或参考不完整、多个响应可接受或评估需要难以编码为字符串匹配的评分标准时，使用 LLM 作为评判者指标（Yu等人，2025a）。响应正确性要求强模型决定答案是否满足用户查询（Deshpande等人，2025），这对开放式响应很方便但引入了评判者依赖性和对提示的敏感性。

忠实度或证据支撑度使用评判者验证声明是否受检索上下文或记忆支持（在某些设置中，提供的引用是否支持响应）（Shao等人，2025；Wang等人，2025f），这有助于区分有用但幻觉的答案与证据支持的答案。

偏好遵循使用评判者确定输出是否尊重显式用户约束或陈述偏好（Zhao等人，2025c；Tavakoli等人，2025；Pakhomov等人，2025），这对个性化基准至关重要，其中正确性由用户对齐而非单一事实标签定义（Wang等人，2024b）。

跨这些指标，一个实际含义是当基准能够将检索或选择与生成分离时，评估变得更可归因于记忆机制（Chen等人，2025a）。检索和排序指标（Recall@K/MAP/NDCG@K）评估记忆接口是否表面正确的项目，而完整性和幻觉导向指标（MI/FMR）暴露可能在累积之前不改变端任务准确率的故障模式（Zhang等人，2025a）。

相比之下，基于相似度的指标在跟踪流畅性和摘要质量方面仍然有用，但它们应与证据支撑检查（FactScore或基于评判者的忠实度）配对，以避免奖励缺乏依据的改写（Aralikatte等人，2021）。此外，一些基准也强调成本和可行性（如容量和效率）（Tan等人，2025a），激励报告 Agent 记住了什么，以及实现该性能所需的计算和存储权衡。

7.2 基准

为评估多样化场景中基础模型 Agent 任务的记忆改进，我们将现有基准分为两个主要领域：用户中心基准和 Agent 中心基准。用户中心基准，如MSC（Xu等人，2022a）和MemoryBank（Zhong等人，2024），主要评估对话一致性，测量 Agent 保留角色信息、回忆用户偏好并跨多会话或多轮对话维持连贯交互的能力。

另一方面，Agent 中心基准，包括OSWorld（Xie等人，2024）和WebArena（Zhou等人，2024），关注记忆在复杂问题解决中的功能应用，测量需要多跳推理和工具使用的任务中的成功率。我们在第7.2.1和7.2.2节中分别介绍常用的用户中心和 Agent 中心基准。

7.2.1 用户中心评估基准

用户中心基准在个性化对话中评估基础模型 Agent，目标是保持与特定用户演化档案和共享交互历史跨长时程的一致性。与 Agent 中心任务（其评估指标在很大程度上是用户不变的（Mo等人，2025））相比，用户中心设置本质上是用户依赖的（Zhao等人，2025d）。Agent 必须决定从对话中存储什么、在需要时检索相关信息、在用户改变偏好时修订信息，并在证据缺失时保持校准（Terranova等人，2025）。

表4总结了具有交互规模（#会话、#问题、最大上下文长度）、数据资源（REAL/SIM/MIX）以及使用 ✓/◐/✗ 的记忆能力和评估覆盖范围的代表性基准。

我们定义十种用户中心记忆能力，用来描述 Agent 在长时程交互中必须记住并使用的内容（Wu等人，2025b；Pakhomov等人，2025）：

- **FE（事实提取）**：从对话中识别并保存可复用的用户属性、约束和关键事件。

- **MR（多会话推理）**：整合分布在多个会话和轮次中的证据。
- **TR（时间推理）**：理解顺序、时间戳和新近度，并在信息变化后选择正确状态。
- **UR（更新与刷新）**：当新证据与旧内容冲突时覆盖过时事实，以最新状态为准。
- **CS（压缩与摘要）**：把长交互历史浓缩为忠实、紧凑且可用的记忆表示。
- **FR（遗忘与保留）**：保留长期有效信息，同时遗忘过时或无关内容以减少干扰。
- **UP（用户事实与偏好）**：跟踪用户角色、偏好、关系、习惯和事件及其变化。
- **AS（助手事实）**：跟踪助手此前的陈述、建议和承诺，保持前后一致。
- **IC（隐式推理与连接）**：连接分散线索，完成多跳或隐式推理。
- **AB（拒答与边界处理）**：识别未知、不可回答、冲突或错误前提，避免在证据不足时捏造答案。

表4中的一个关键模式是MR和UP是最一致覆盖的能力，反映了用户记忆的主导框架是从长对话中检索用户相关事实并在后续使用。相比之下，操作能力的定义远不统一。早期的长对话基准如MSC（Xu等人，2022a）和DuLeMon（Xu等人，2022b）强调多会话连贯性和角色使用，但其评估严重依赖基于相似度或基于生成的指标（如BLEU、Distinct-n、困惑度、ROUGE），这弱隔离了记忆。

流畅性和通用帮助性可能掩盖不正确的回忆，使FE、UR和FR难以归因。较新的基准越来越多地通过定义特定任务类型或评估来转向机制可归因评估。MemoryBank（Zhong等人，2024）和HaluMem（Chen等人，2025a）引入显式记忆记录和操作级搜索，将提取、更新和记忆完整性失败转化为可测量目标。

LongMemEval（Wu等人，2025b）通过根据记忆能力显式分类问题类型进一步加强归因，支持更清晰地定位时间推理、更新或回避行为的失败。然而，CS和FR在当前的基准设计中仍然相对评估不足且系统性评估较少。压缩仅在少数基准中显式出现（Hu等人，2025c），选择性遗忘或保留（Jia等人，2025b）经常是部分的或缺失的，尽管对于在有限记忆预算和演化用户状态下运作的长时程助手至关重要。

AB也不一致地需要（Ai等人，2025；Jiang等人，2025a）。只有少数基准显式奖励证据缺失下的回避，留下评估防止自信幻觉的安全记忆行为的空白。

表4中的评估遵循几个主要方面，反映基准提供的评估程度。

首先，许多基准把评估简化为答案级正确性，并报告准确率或 F1，例如 PerLTQA、LoCoMo、DialSim、PersonaMem 和 MemoryAgentBench。

当任务要求选择支持性记忆而不是生成自由文本时，研究通常采用 Recall@K、MAP 或 NDCG@K 等检索与排序指标，例如 PerLTQA 和 LongMemEval。其次，一些用户中心基准开始越过最终答案，直接评估记忆系统的行为。

HaluMem报告记忆准确率和完整性以及虚假记忆率，以量化幻觉或不正确记忆操作（Chen等人，2025a），LOCOO（Jia等人，2025b）报告记忆保留分数（MRS），MemBench（Tan等人，2025a）显式测量容量和效率以捕捉固定记忆预算下的性能和成本权衡。

第三，当输出是开放式的或参考答案不足时，基准越来越多地依赖 LLM 作为评判者来评分响应正确性或偏好遵循（例如，LongMemEval（Wu等人，2025b）、PrefEval（Zhao等人，2025c）和MemoryBench（Ai等人，2025））。与精确匹配评分相比，基于评判者的评估扩展了对现实助手行为的覆盖，但对评估器模型和评分标准更敏感。

表4：用户中心记忆基准概览。记忆能力——FE：事实提取；MR：多会话推理；TR：时间推理；UR：更新与刷新；CS：压缩与摘要；FR：遗忘与保留；UP：用户事实与偏好；AS：助手事实；IC：隐式推理与连接；AB：回避与边界处理。标记——✓：显式覆盖。基准将此能力定义为具有专用任务类型或注释的目标并直接评估；◐：部分或间接覆盖。能力可能在某些实例中被要求或由设置隐含，但缺乏专用标签或能力特定评估，因此覆盖较弱或不可干净归因；✗：未覆盖。

不存在对应任务、注释或评分标准，评估不要求或测量此能力。资源——REAL（人工撰写或真实世界对话）、SIM（完全合成或模拟）、MIX（真实和合成的混合）。评估指标——AR：准确检索；TTL：测试时学习；LRU：长距离理解；SF：选择性遗忘；MM-R：多消息相关性；RC：响应正确性；CC：上下文连贯性；MA：记忆准确率；MI：记忆完整性；FMR：虚假记忆率；MRS：记忆保留分数；JUDGE：LLM 作为评判者。

名称	#会话	#问题	最大令牌	FE	MR	TR	UR	CS	FR	UP	AS	IC	AB	资源	链接	评估
MSC	5K	-	1K	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	REAL	-	困惑度
DuLeMon	27,501	-	1K	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	REAL	-	F1, Recall@K, BLEU, Distinct-n
MemoryBank	300	194	5K	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	SIM	-	检索准确率, RC, CC
PerLTQA	3,409	8,593	1M	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	SIM	-	准确率, F1, Recall@K, MAP
LoCoMo	1K	7,512	10K	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	MIX	-	F1, ROUGE, BLEU, MM-R

名称	#会话	#问题	最大令牌	FE	MR	TR	UR	CS	FR	UP	AS	IC	AB	资源	链接	评估
DialSim	~1,300	1M	367K	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗	MIX	-	准确率
LOCOCO	3,080	2,981	-	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	SIM	-	准确率, MRS
MemoryAgentBench	130	207	1.44M	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	SIM	-	准确率, F1, AR, TTL, LRU, SF
LongMemEval	50K	500	1.5M	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	MIX	-	JUDGE, Recall@K, NDCG@K
HaluMem	1,387	3,467	1M	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	MIX	-	MA, MI, FMR
PersonaMem	60	~6K	1M	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	SIM	-	准确率
PrefEval	-	3,000	100K	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	MIX	-	JUDGE, 准确率
MemBench	65K	53K	100K	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	MIX	-	准确率, 容量, 效率
MemoryBench	20K	-	-	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	MIX	-	准确率, F1, JUDGE
ConvoMem	300	75,335	3M	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	SIM	-	准确率

7.2.2 Agent 中心评估基准

Agent 中心基准评估基础模型作为半自主 Agent，必须在环境中执行多步动作以达到明确指定的目标，而非仅对静态提示产生响应。这些基准通常提供任务规范（如初始状态及约束）以及目标成功检查器，并用端到端指标如文本QA任务的准确率/F1或交互环境的成功率（SR）来总结性能。

因为正确性由环境状态定义，目标结果在很大程度上是用户不变的：在相同任务设置下，不同用户发出相同请求应获得相同的完成信号，即使 Agent 遵循不同的轨迹（Trivedi等人，2024；Zhou等人，2024）。

我们在表5中总结了常用的 Agent 中心基准。除了按任务环境（Env）和接口（Interact）标记每个基准外，我们进一步注释（i）资源类型，指示任务世界和证据是用真实世界还是模拟数据构建，或两者的混合，以及（ii）基准最直接锻炼的核心 Agent 能力（Abilities）。

常用核心能力标签包括：TEMP（时间与顺序推理）、STATE（环境和任务状态跟踪）、GROUND（指令-环境对齐）、PLAN（规划与重规划）、TOOL（工具调用）、MHOP（多跳推理）和 DIAL（目标导向对话管理）。

软件工程任务还使用 DEBUG、CODEGEN 和 PATCH，分别表示故障诊断、代码生成和代码修复。TTL 表示测试时学习，即 Agent 不更新参数，而是通过记忆积累经验来改善后续表现。

跨环境的 Agent 中心评估涵盖 TEXT/WEB 信息检索、OS/APP 计算机使用、CODE 软件工程、ROBOT/GAME 具身控制，以及 VIDEO/PAPER 长内容工作流。

这些环境引发不同的记忆压力。文本和多跳QA强调中间事实的证据跟踪和状态记录（Ho等人，2020；Trivedi等人，2022）。Web、桌面和应用程序设置强调情景动作记忆。

基础模型 Agent 必须记住访问过的页面、填写的字段、下载的文件、先前的工具输出、交互过程中发现的约束，并避免冗余探索（Deng等人，2023；Zhou等人，2024；Yao等人，2022）。代码和论文工作流需要工作记忆，如文件、补丁、假设和实验日志，以支持跨阶段连续性（Starace等人，2025；Miao等人，2025）。这些要求使压缩、选择和检索对完成任务至关重要（Jimenez等人，2024；Qiu等人，2025a）。

视频基准额外测试跨长剪辑的感觉记忆，其中关键线索可能在问题提出之前很久出现（Fu等人，2025；Wu等人，2024a）。随着基准变得更加现实和长时程，基础模型 Agent 无法在提示中保留所有任务相关上下文，必须通过显式记忆操作将状态外化，以随时间保留和检索关键信息。

表 5 中基准的评估方法主要分为几类。TEXT 的答案级准确率/F1（Trivedi 等人，2022；Phan 等人，2025）、交互式 WEB/OS/APP/ROBOT/GAME 的基于目标的SR、GC（Xie 等人，2024；Trivedi 等人，2024；Shridhar 等人，2020；2021；Fan 等人，2022），以及CODE的执行中心指标（如Pass@1、RR）（Jimenez等人，2024；Qiu等人，2025a）。

对于记忆中心分析，两个额外维度至关重要：（1）依赖距离——所需信息与其后续使用之间距离多远，如轮内、跨轮或跨会话，以及（2）交互下的记忆正确性——存储项目是否保持忠实、非矛盾且随环境演化策略一致。这激励用记忆敏感测量补充端到端成功，如：（i）所需事实或工具输出的检索忠实度和覆盖率，（ii）状态跟踪中的错误模式（漂移、遗漏、矛盾），（iii）中断下的持久性（长间隔后恢复），以及（iv）效率权衡（记忆大小、更新频率和检索成本）。

我们期望下一代基准超越单一的端到端准确率或成功率，转而纳入记忆相关指标，使评估也能归因于记忆机制而非短时程提示或偶然启发式。

表5：Agent 中心任务基准概览（§7.2.2）。每个基准按任务环境（Env）、交互模式（Interact）、数据源（Resource）、核心能力（Abilities）和评估指标（Evaluation）注释。Env - TEXT：文本/文档；WEB：Web；OS：操作系统/桌面；APP：应用/API中心；

CODE：代码/软件工程；ROBOT：具身机器人；GAME：游戏/模拟；VIDEO：视频（长格式）；PAPER：科学论文/研究。

Interact - QA：问答；MT：多轮；GUI：图形UI；API：工具/API调用；EXEC：基于执行；MM：多模态；ACT：动作/控制。Resource - REAL：真实世界；SIM：模拟/合成；MIX：真实+模拟混合。

Abilities - MHOP：多跳推理；PLAN：规划/行动；STATE：状态跟踪；GROUND：指令-环境对齐；TOOL：工具使用；DEBUG：调试；CODEGEN：代码生成；PATCH：补丁/修复；TEMP：时间推理；DIAL：对话管理；TTL：测试时学习。Evaluation - SR：成功率/解决率；PR：通过率；GC：目标完成；RR：解决率；LCCS：最长连续正确序列。

名称	#数据	环境	交互	资源	核心能力	链接	评估
HotpotQA	113K	TEXT	QA	REAL	MHOP, STATE	Github	准确率, F1
2WikiMultiHopQA	193K	TEXT	QA	REAL	MHOP, STATE	Github	准确率, F1
MuSiQue	25K	TEXT	QA	REAL	MHOP, STATE	Github	F1
HLE	2.5K	TEXT	QA	REAL	MHOP, STATE	网站	准确率, RMSE
BrowseComp	1,266	WEB	QA, GUI	REAL	PLAN, TOOL, MHOP, STATE	网站	准确率, PR
Mind2Web	2.35K	WEB	GUI	REAL	GROUND, PLAN, STATE	网站	准确率, F1, SR
WebArena	812	WEB	GUI	SIM	GROUND, PLAN, STATE	网站	SR
WebShop	12.1K	WEB	GUI	MIX	GROUND, PLAN, STATE	Github	任务分数, SR
GAIA	466	WEB	QA, GUI	REAL	TOOL, MHOP, PLAN, STATE	网站	准确率, SR
OSWorld	369	OS	GUI, MM	REAL	GROUND, PLAN, STATE	网站	SR
AppWorld	750	APP	API, MT	SIM	TOOL, CODEGEN, PLAN, STATE	网站	SR
τ-Bench	165	APP	API, MT	SIM	TOOL, PLAN, STATE, DIAL	Github	Pass^1, Pass^k
τ²-Bench	2.3K	APP	API, MT	SIM	TOOL, PLAN, STATE, DIAL	Github	Pass^1, Pass^k
HumanEval	164	CODE	EXEC	REAL	CODEGEN, DEBUG	Github	Pass@1
SWE-Bench	2.3K	CODE	EXEC	REAL	PATCH, DEBUG, STATE	网站	RR
LoCoBench	8K	CODE	EXEC	SIM	CODEGEN, PATCH, DEBUG, STATE	Github	多指标
LoCoBench-Agent	8K	CODE	EXEC	SIM	TOOL, PLAN, CODEGEN, PATCH, DEBUG, STATE	Github	多指标
PaperBench	20	PAPER	EXEC, MT	REAL	TOOL, PLAN, CODEGEN, DEBUG, STATE	网站	复制分数
RECODE-H	102	PAPER	EXEC, MT	REAL	CODEGEN, PATCH, DEBUG, DIAL, STATE	Github	召回率, PR
ALFRED	25.7K	ROBOT	ACT, MT	SIM	GROUND, PLAN, STATE	网站	SR, GC
ALFWorld	3.8K	ROBOT	ACT, MT	SIM	PLAN, STATE, GROUND	网站	SR
MineDojo	3.1K	GAME	ACT, MT	MIX	PLAN, STATE, GROUND	网站	SR
EgoSchema	5,031	VIDEO	QA, MM	REAL	TEMP	网站	准确率
Video-MME	2,700	VIDEO	QA, MM	REAL	TEMP	网站	准确率
LongVideoBench	6.7K	VIDEO	QA, MM	REAL	TEMP	网站	准确率
MT-Mind2Web	720	WEB	GUI, MT	REAL	GROUND, PLAN, STATE, DIAL	Github	准确率, F1, SR
MPR	10.8K	TEXT	QA	SIM	MHOP, STATE	Github	准确率
StoryBench	397	GAME	MT, ACT	SIM	PLAN, STATE, TEMP	-	准确率, LCCS
Evo-Memory	~3,700	TEXT	QA, MT	MIX	TTL, PLAN, STATE	-	准确率, SR
LifelongAgentBench	1,396	APP, OS	API, MT	SIM	TTL, TOOL, PLAN, STATE	网站	SR
OdysseyBench	602	APP	GUI, MT	MIX	PLAN, TOOL, STATE, TEMP	Github	PR

8 应用

记忆将 LLM 转变为动态的、持久的 Agent，代表了近期研究的根本性转变。当在复杂现实场景中实现时，Agent 记忆不仅作为存储工具出现，而且作为认知载体，使连续性、学习和个性化成为可能，连接 Agent 的过往经验与其未来行动。近期工作已广泛研究了 LLM Agent 中记忆启用的能力，其中记忆的存储、操作和管理方式差异显著。

为提供关于记忆设计改进如何促进进一步能力的见解，本节讨论和总结了跨教育、科学研究、游戏和模拟、机器人、医疗保健、对话系统、工作流自动化、软件工程、在线流媒体和推荐、信息搜索、金融和会计以及法律和咨询等领域的近期代表性工作。应用领域如图8所示，总结见表6。

图8：基础模型 Agent 记忆的主要应用领域，包括教育、科研、游戏与模拟、机器人、医疗、对话系统、工作流自动化、软件工程、信息搜索、金融、法律咨询和流式媒体推荐。

教育。教育 Agent 需要跨长时段持续的个性化交互，使记忆对于跟踪学习者进度、适应教学和维持教学连贯性至关重要（Chu等人，2025）。没有记忆，Agent 将每次交互视为孤立事件，无法建立在学生先前知识基础上或维持教学一致性。最近的模型说明了向更复杂记忆模块的转变。例如，LOOM（Cui等人，2025a）利用学习者记忆图映射具有先决依赖关系的教育概念，以促进个性化课程生成。

Agent4Edu（Gao等人，2025b）依据艾宾浩斯遗忘曲线，建模教师培训中的知识衰减。WebCoach（Liu等人，2025b）使用持久跨会话记忆支持自我进化的教学指导。这些系统揭示了在教育领域，记忆的功能更少是历史日志，更多是学生的认知数字孪生（Zheng等人，2022）。未来工作应朝向可互操作的记忆协议，允许学习者的认知档案跨不同教育平台持久化，有效创建其智力发展的演化记录。

科学研究。科学研究代表了一个前沿领域，其过程漫长且昂贵，要求 Agent 综合大量文献、管理来源并在多阶段工作中维持推理连续性。在近期研究中，General Agentic Memory（GAM）（Yan等人，2025a）为深度研究采用专门的研究 Agent，通过通用页面存储支持复杂多跳推理的动态上下文重构。IterResearch（Chen等人，2025c）维护仅保留演化报告和即时结果的工作空间以防止上下文过载。

MirrorMind（Zeng等人，2025）通过检索特定认知风格和知识库的分层架构模拟集体智能。AISAC（Bhattacharya & Som，2025）实现结合语义检索与结构化SQLite日志以确保可重复性的混合记忆。这些研究 Agent 例证了一个范式转变，其中记忆作为研究过程的验证层，维护结论如何达成的透明谱系。

未来系统将从单体研究助手演变为实验室规模的集体智能，其中多个 Agent 共享特定科学领域的统一、演化知识图谱，随新论文的发表和综合实时更新。

游戏与模拟。在开放世界游戏环境和模拟中，记忆支持技能获取、空间探索和复杂社会动态的出现。Agent 必须利用程序性记忆保留学到的技能，利用情景记忆维持可信的社会历史。例如，Voyager（Wang等人，2025c）将成功动作存储为技能库中的可执行代码，以实现能力组合。GITM（Zhu等人，2023）采用分层文本基记忆，其中规划器记录结构化目标并基于相似性检索相关经验。

Generative Agents（Park等人，2023）开创了情景记忆流，Agent 通过计划、反思和反应维持可信的个人历史。GameGPT（Chen等人，2023）使用多 Agent 协作，其中记忆缓冲区分游戏开发生命周期的不同阶段。M2PA（Zhou等人，2025b）受认知理论启发，采用多记忆规划 Agent，记忆选择由感觉、工作和情景记忆之间的交互决定。

这些应用突显了记忆使模拟 Agent 能够展现涌现的个性化行为，超越了脚本交互。未来方向包括压缩长时间游戏会话的程序性技能和使虚拟角色能够形成跨越多个模拟事件的持久社会关系。

机器人。具身 Agent 在部分可观察的物理环境中运作，需要记忆来弥合高层推理与低层控制之间的差距。记忆对于维持空间理解、跟踪目标进度和从过去的试错中学习至关重要。Memo（Gupta等人，2025）侧重于在强化学习训练期间降低具身 Agent 中的记忆开销，而MG-Nav（Wang等人，2025b）构建双尺度空间记忆以改善视觉导航。

JARVIS-1（Wang等人，2024p）在开放世界环境中通过多模态记忆实现终身学习，KARMA（Wang等人，2025t）将短期情节记忆与长期语义记忆集成以完成复杂具身任务。这些系统表明机器人的记忆不仅仅是记录感觉读数，而是构建用于推理的动作-结果模型。未来工作应开

发统一的“具身记忆”表示，将空间图、对象持久性和因果依赖关系集成为一个连贯的、可查询的结构，使机器人能够推理物理世界中的长期效应。

医疗保健。医疗 Agent 需要记忆来维护生理趋势和情感状态的纵向记录，这对于建立用户信任和依从性至关重要。TheraMind（Hu等人，2025a）为心理辅导采用分层记忆系统，跟踪患者的情绪演变和治疗里程碑。DAM（Lu & Li，2025）专注于动态情感记忆管理，根据对话上下文调整情感基调。Mem-PAL（Huang等人，2025d）为长期患者-Agent 交互构建个性化记忆。

在手术AI中，ReSurgSAM2（Liu等人，2025d）使用长期视频记忆跟踪手术场景中的工具和解剖结构。这些医疗应用强调记忆不仅必须存储事实，还必须建模随时间的情感动态。主要挑战是确保医疗记忆的隐私和对抗性鲁棒性——患者的敏感数据必须保持加密，同时允许 Agent 进行推理。未来系统可能实施联合记忆架构，其中模型在分散的数据上本地更新，而汇总洞察贡献于全球医学知识库。

对话系统。作为记忆增强最直接的应用，对话 Agent 必须管理上下文窗口约束、维持角色一致性，并模拟持久的类似人类关系。A-Mem（Xu等人，2025e）将记忆管理视为 Agent 行为，提供读取、写入和反思操作的明确策略。MemGPT（Packer等人，2023）将 LLM 概念化为操作系统，具有显式记忆层级进行上下文管理。O-Mem（Wang等人，2025i）实现了全向记忆，统一短期和长期信息。

MemoChat（Lu等人，2023）微调 LLM 以有效使用备忘录进行一致的长程开放域对话。这些对话系统表明，有效的长期对话不仅依赖于记忆什么，还依赖于遗忘什么。未来工作应优先考虑“对话记忆优化”——开发能够在信息保留、个性化和隐私保护之间动态平衡的 Agent，本质上是使 Agent 成为熟练的对话者，而非被动的记录员。

工作流自动化。用于工作流自动化的 Agent 必须将重复的多步流程归纳为可重用的模板，并随时间适应界面变化。记忆使这些 Agent 能够学习工具使用模式并在长时间运行的企业环境中保持状态一致性。Agent Workflow Memory（AWM）（Wang等人，2025v）显式编码程序性工作流记忆，实现跨任务自动化。ToolMem（Xiao等人，2025）引入了一个具有可学习工具能力记忆的多模态框架。

Synapse（Zheng等人，2024）使用轨迹即示例提示，通过利用过去成功执行的经验记忆来改善计算机控制。此外，WebArena（Zhou等人，2024）提供了一个测试平台，其中通过跨站点记忆的长期导航取得成功。这些应用强调了工作流自动化中的记忆从简单宏记录器演变为理解任务依赖和用户意图的主动编排器。未来框架应该能够在流程偏差发生时检测和纠正记忆漂移，确保企业 Agent 保持可靠。

软件工程。对于软件工程，Agent 在需要长时程推理、多文件协调和积累调试经验的复杂代码库中运作。记忆使这些 Agent 能够维护代码上下文、从实现尝试中学习并导航大型仓库。MetaGPT（Hong等人，2024）为开发工作流实现程序性记忆，同时维护项目规范的共享语义记忆。ChatDev（Qian等人，2024a）通过开发迭代的情节记忆扩展了这一能力，以从调试会话中学习。

SWE-bench（Jimenez等人，2024）上的评估报告记忆机制通过维护跨多文件编辑的上下文和利用先前调试经验来改善问题解决。关键地，有效的编码 Agent 不仅生成代码；它们回忆先前失败和修复的轨迹，以在通过单元测试方面实现更高成功率。该领域的未来应用可能超越本地项目记忆，转向共享的、匿名化的知识库，其中分布式编码 Agent 贡献并查询算法解决方案和错误修复方案的共享知识库，加速全球自动化软件开发的速度。

流式媒体与推荐。在在线流媒体和推荐系统时代，Agent 处理高通量多模态输入，其中信息的相关性随时间动态变化。记忆允许 Agent 维持时间一致性并识别跨视频帧和用户交互的长距离模式。例如，WorldMM（Yeo等人，2025b）利用动态多模态记忆存储视觉-语言特征，以在长时间视频流上进行复杂推理。GCAgent（Yeo等人，2025a）引入双结构情节记忆，将模式知识与叙述序列分离，用于结构化视频理解。

类似地，Xiong等人（2025b）实现了支持具有保留上下文的多轮交互的记忆增强知识缓冲区。这些应用表明，在流媒体上下文中，记忆充当将瞬态数据蒸馏为持久表示的时间过滤器。未来研究应关注遗忘感知推荐记忆，能够区分用户短暂兴趣与长期偏好，优化实时推送中新颖性与相关性之间的平衡。

信息搜索。超越简单检索，信息搜索 Agent 必须综合冲突报告、跟踪演化故事并管理大量文档空间而不损失推理深度。记忆充当将静态搜索结果转换为知识合成活动工作空间的组织框架。AgentFold（Ye等人，2025b）通过主动上下文管理解决长时程Web导航问题，折叠无关轨迹以防止溢出同时保留关键发现。MemSearcher（Yuan等人，2025a）采用强化学习进行联合搜索和记忆管理。

MoM（Zhao等人，2025a）利用场景感知记忆，将查询动态路由到专门的记忆库。此外，Rajesh等人（2025）将RAG与情景记忆桥接，维护搜索过程本身的存储库。这些系统表明有效搜索不仅是找到数据，而是管理搜索轨迹的认知负载。下一代信息搜索 Agent 可能朝向协作记忆结构演化，其中多个 Agent 验证事实并在应对突发新闻周期时更新共享信念图。

金融与会计。金融领域以高频波动、定量数据和定性新闻的混合以及长期战略一致性的关键需求为特征。记忆在此至关重要，因为交易和会计要求 Agent 处理实时信号并回忆历史市场制度，同时维持持久交易角色以避免不稳定决策。近期工作为此目的引入了专门架构。FinMem（Yu等人，2025e）实现分层记忆系统，将即时市场观察与长期投资经验分离，允许 Agent 随时间精炼其个性和风险档案。

FinCon（Yu等人，2024a）利用多 Agent 设置，其中记忆作为概念性言语强化的存储库，使系统能够通过反思性反馈循环从过去的财务决策中学习。QuantAgent（Wang等人，2024g）进一步通过自我改进机制寻求投资轨迹，其中成功轨迹作为程序性记忆存储以供未来策略精炼。尽管取得这些进展，主要挑战仍然是金融记忆中的信噪比。Agent 必须学会区分瞬态市场波动与根本性转变。

未来研究应探索遗忘感知金融 Agent，能够修剪过时的经济假设同时保留核心风险管理原则。

法律与咨询。在法律和咨询服务中，Agent 必须导航大量异构文档，其中每个声明的精确来源是强制性的。记忆是允许这些 Agent 对长期案件进行多步推理的认知载体，确保建议与先前引用的法规或客户历史保持一致。例如，MALR（Yuan等人，2024b）利用多 Agent 框架通过维护模拟法律专家之间协作辩论的交互历史来改善复杂法律推理。

StaffPro（Maritan，2025）聚焦咨询方面，通过使用记忆随时间分析工作人员和项目需求，通过过往绩效数据的反馈循环实现动态人员配置。Blair-Stanek等人（2025）展示了记忆在发现新型税收最小化策略中的力量，通过将数千页演化法规和判例法综合为持久推理图。该领域的核心挑战是幻觉记忆的高风险。单个记错的条款可能导致法律责任。

未来方向可能聚焦于可验证记忆架构，将每个检索到的洞察链接回加密签名的源文档，确保最高水平的专业诚信和问责制。

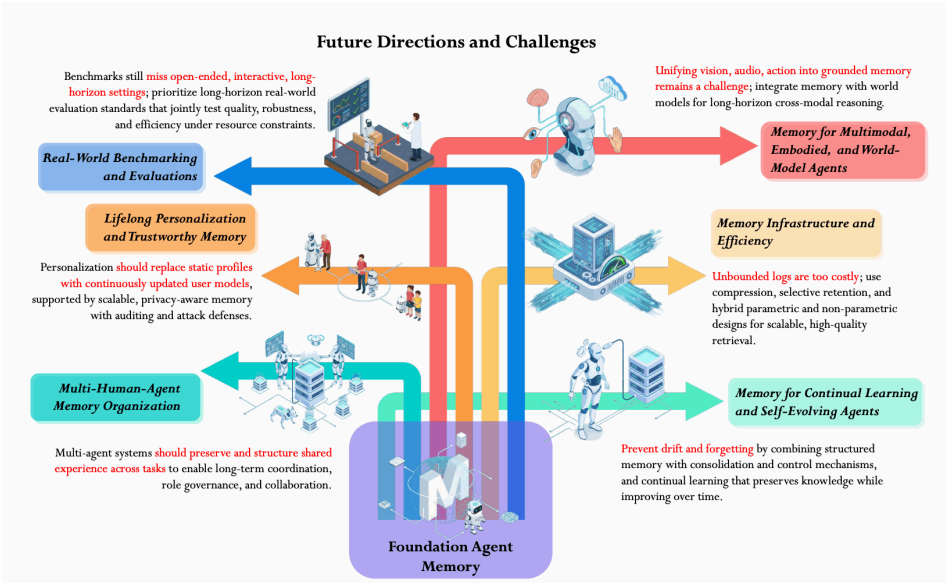

图9：基础模型 Agent 记忆的未来方向与挑战，包括持续学习和自我进化、多用户与多 Agent 记忆组织、记忆基础设施与效率、终身个性化与可信记忆、多模态具身及世界模型 Agent，以及现实世界基准测试。

9 未来方向

基于前述分析，我们强调六个我们认为对基础模型 Agent 记忆最为重要的开放挑战。图9总结了它们以及我们对每个方向认为最有前景的方向。

9.1 持续学习和自我进化 Agent 的记忆

记忆启用的自我进化 Agent 中的一个基本挑战在于跨任务内和跨任务时间尺度管理记忆动态。在任务内层面，Agent 必须在严格的上下文窗口约束下持续决定从异构流（如工具输出、搜索结果、反馈信号和中间推理轨迹）中保留、压缩或丢弃什么信息（Gao等人，2025a）。现有系统很大程度上依赖启发式记忆控制器，使记忆进化与推理行为之间的内部耦合关系理解不足。

在跨任务层面，Agent 预期跨事件和任务分布积累经验（Wei等人，2025d），但当前方法主要强调推理时重用而非原则性巩固和泛化。相比之下，经典持续学习方法侧重于通过重放、正则化或参数隔离来防止灾难性遗忘（Rebuffi等人，2017；Lopez-Paz & Ranzato，2017；Shin等人，2017；Kemker & Kanan，2018），但它们通常将记忆视为知识保留的静态机制。

这种框架对于基于 Agent 的系统来说是不够的，其中记忆还必须跟踪演化交互状态、用户特定信息和程序性行为。此外，从积累经验中进行稳定的后训练适应仍未得到充分探索，存在负迁移、不受控漂移和语义不一致的未解决风险（Ke等人，2025b）。

未来研究应因此将持续学习重新框定在更丰富的、Agent 中心的记忆概念周围，整合语义、情景和程序性组件（Ke等人，2025a；Sumers等人，2023）。超越显式文本日志，潜在和结构化记忆表示为可扩展和高效适应提供了有前景的方向，支持紧凑存储同时保留因果和行为抽象。进展很可能需要超越推理时启发式，转向利用积累的 Agent 经验进行持续改进的原则性后训练范式。

这包括设计选择性地蒸馏长期知识、将演化记忆与模型参数对齐并减轻遗忘而不牺牲可塑性的巩固机制。相应地，需要新的基准来评估不仅任务级保留，还包括持续适应、相关性感知记忆管理和非平稳目标和环境下的行为稳定性（Ke等人，2024）。建立将经典持续学习目标与结构化记忆设计连接起来的统一框架仍然是自我进化基础模型 Agent 的一个关键开放方向。

9.2 多用户与多 Agent 的记忆组织

最近的多 Agent LLM 框架，如AutoGen（Wu等人，2024b）和 AgentLite（Liu等人，2024c），通过结构化消息传递和提示驱动控制支持任务分解和基于角色的协调。

在实践中，这些系统日益在人类-Agent 协作设置中运作，其中AI Agent 不仅与其他 Agent 交互，还与人类用户或监督者通过迭代反馈、纠正和委派交互（Lu等人，2025c；Zou等人，2025b；c），基准开始在可配置人类参与和显式角色与权限结构下评估此类系统（Wu等人，2026）。

然而，尽管复杂性不断增长，协调在很大程度上仍然是情景和瞬态的：交互限定于单个任务实例，一旦任务完成，几乎不留存经验（Suzgun等人，2025）。因此，Agent-Agent 和人类-Agent 协作在每次任务中都被从头重新建立，限制了系统适应交互策略、个性化行为或在重复任务或部署中改善协作质量的能力。

使交互实体（包括基础模型 Agent 和人类）之间能够进行持久和自适应协作将激发长期研究问题（Han等人，2024；Li等人，2024c）。一个重要方向是协作（社会）记忆，其中 Agent 保留关于其协作者的经验，如通信偏好、领域专业知识、反馈模式或历史交互结果，使其能够适应信号策略、校准信任并随时间减少协调开销。

同时，Agent 可能受益于角色特定工作流和程序性记忆，积累关于自身重复工作流的经验（Wang等人，2025v），包括任务分解模式、执行策略和常见故障模式，使承担稳定功能角色的 Agent 能够通过经验驱动的专业化逐步精炼其行为。在此类多实体设置中引入持久记忆也引发了记忆治理和协调挑战，包括所有权、访问、责任以及如何处理不同视角或人类纠正等问题。

解决这些问题对于防止不受控错误传播和在多 Agent 系统规模、异构性和任务复杂性增长时维持可靠协作至关重要。

9.3 记忆基础设施与效率

随着基础模型 Agent 越来越多地部署在长时程、交互式和开放世界环境中，记忆基础设施已成为核心效率瓶颈（Chhikara等人，2025；Qiu等人，2025b）。大多数现有 Agent 记忆设计仍然是文本中心的，将记忆视为不断增长的过往交互、摘要或情景日志的集合，这些被检索并注入提示中（Xu等人，2025e；Zhong等人，2024）。

虽然这种策略可以改善任务性能，但它引入了大量 token 开销，记忆上下文通常扩展到数千个 token 并表现出递减的边际回报（Chhikara等人，2025）。记忆成本的这种线性增长（Kwon等人，2023）直接转化为更高的推理延迟和降低的可扩展性，特别是在多轮或终身设置中。更根本地，当前方法将记忆容量与提示长度混为一谈，隐含假设更多上下文意味着更好推理。这种假设忽视了选择性保留、结构化访问和经验巩固的需求。

此外，记忆通常通过启发式方法如总结或截断在外部管理，而非集成到 Agent 的推理和学习过程中。这些局限性突显了一个核心挑战：如何设计记忆系统使 Agent 能够在严格资源约束下高效保留、抽象和重用经验，而不依赖无限的上下文扩展。

未来关于记忆基础设施和效率的研究可以被视为朝向日益抽象和集成的经验表示的过程。在近期，一个前景方向在于有组织的文本基记忆，其中文本记忆被显式结构化以实现高效访问而非最大覆盖。近期工作已探索基于模式或图结构的记忆表示（Edge等人，2024；Chhikara等人，2025），但这些努力主要针对推理准确性而非效率。一个开放机遇是设计结构感知存储和精度导向检索机制，仅暴露推理关键跨度，最小化不必要的上下文注入。

超越文本组织，效率增益可以通过压缩隐状态记忆实现，其中情景、语义性或程序性经验被编码为作为持久记忆单元而非仅相似度索引的紧凑向量表示。在更深层次的集成水平上，内化或参数化记忆提供了通往恒定大小记忆的路径，其中长期经验被吸收到内部状态或模型参数中。

诸如MEM1（Zhou等人，2025c）和Mem-α（Wang等人，2025q）等框架通过强化学习训练 Agent 将记忆巩固、更新和丢弃作为推理过程本身的一部分来例证此转变，即使在长时程任务中也能实现有界记忆。实现这些方向还将需要能够支持受控、多步交互和可扩展评估的稳健环境基础设施。诸如NeMo Gym（NVIDIA，2025）等平台将环境逻辑与训练解耦并提供模块化奖励和验证服务，代表了此生态系统的重要组成部分。

共同地，这些进展表明了一个未来，其中记忆不再是外部提示管理工件，而是与 Agent 推理和决策共同进化的核心学习子系统，通过集成记忆架构实现，该架构结合了结构化隐状态表示（如具有可微读写接口的分层向量表）、通过端到端强化学习或元学习目标对记忆和策略的联合优化，以及基于任务相关性、不确定性估计和长期效用的动态分配、压缩和退役记忆单元的自适应记忆控制器。

9.4 终身个性化与可信记忆

终身个性化旨在使基础模型 Agent 具备跨会话、任务和长时间跨度持续适应个体用户的能力（Wang等人，2024i）。与依赖静态用户档案或瞬态上下文信号的传统个性化方法不同，此设置要求 Agent 维护捕捉逐步偏好漂移、长期目标和行为规律的演化用户表示。

虽然近期关于持久记忆和动态用户建模的努力已取得初步进展（Zhong等人，2024；Tan等人，2025c；Zhang等人，2025o），但现有系统很大程度上依赖交互历史的启发式聚合或非结构化记忆检索，这限制了它们蒸馏可靠、可解释和具有因果依据的用户知识的能力（Pink等人，2025）。

此外，长时程个性化在记忆陈旧性、概念漂移和信用分配方面引入了复杂挑战：Agent 必须决定哪些过往交互保持相关、如何随时间协调冲突信号以及如何防止过时偏好主导当前行为。这些问题因可扩展性约束而进一步加剧，因为直接保留或重放长交互历史导致在现实世界、始终在线的助手设置中部署时产生高昂的存储、检索和推理成本。

一个关键研究方向是设计可扩展和动态记忆系统，能够增量更新用户建模，同时将细粒度情景轨迹与高层抽象（如偏好、习惯或长期意图）桥接。有前景的方法包括将短期情景缓冲区与蒸馏语义用户档案分离的分层记忆架构（Tan等人，2025c）、调节何时写入、压缩或覆盖用户信息的学习记忆控制器（Zhang等人，2025o），以及在分布漂移下减轻遗忘的持续表示学习技术（De Lange等人，2021；Parisi等人，2019）。

同时，该领域需要为终身个性化量身定制的新评估基准，超越单轮准确率，转向评估长期一致性、对偏好变化的适应性以及扩展交互下的鲁棒性（Xu等人，2025e）。同样重要的是可信记忆基础设施的发展。持久用户记忆带来了重大风险，包括隐私泄露（Wang等人，2025a）、记忆投毒（Tan等人，2024b）和对抗性操纵（Dong等人，2025），这些可能随时间无声积累。

近期关于安全和可审计记忆模块的工作（Wei等人，2025b；Wang等人，2025a）突显了对支持存储记忆的检查、编辑和撤销的用户可控机制的需求，以及对未授权访问和恶意写入的防御。最终，鲁棒性、透明度和安全性应被视为与适应性同等重要的一级设计目标，在设计评估终身个性化基础模型 Agent 时（Yu等人，2025c）。

9.5 多模态、具身和世界模型 Agent 的记忆

下一代基础模型 Agent 的核心挑战在于设计能够忠实表示、对齐和抽象异构感觉流（包括视觉、音频、语言、触觉反馈和本体感觉信号）为连贯内部状态的记忆系统（Bei等人，2026）。虽然文本记忆机制在长时程推理和个性化方面已取得显著成功（Xu等人，2025e），但现有

方法在很大程度上假设单模态或语言主导表示。

多模态 Agent 记忆的早期努力（Long等人，2025；Bo等人，2025；Liu等人，2025g）揭示，直接将文本基记忆扩展到高维感知输入会导致严重低效、跨模态语义错位和脆弱检索行为。这些挑战在具身设置中进一步加剧，Agent 在闭环环境中运作，必须推理时间扩展的感知-行动-结果轨迹。在此类场景中，记忆必须超越存储情景观察，转而编码与环境动力学、可供性和物理约束相结合的知识（Wang等人，2025t）。

然而，当前系统缺乏用于动作条件记忆更新、跨模态抽象和跨情景、语义性和程序性记忆层一致性维护的原则性机制。因此，具身 Agent 经常遭受技能碎片化、长时程规划失败以及由错位或陈旧记忆引起的累积错误。

展望未来，一个有前景的研究方向是将 Agent 记忆提升为显式的、预测性世界模型，将记忆视为随时间的可控内部状态，而非被动日志。世界模型的建模方式（Hafner等人，2023；Ha & Schmidhuber，2018）提供了统一视角，其中记忆更新可以被建模为依赖于感知和行动的隐状态转换。

这为主动记忆规划打开了大门，其中 Agent 在提交更新之前模拟存储、压缩或遗忘信息的长期后果（Schrittwieser等人，2020；Silver等人，2017）。在此框架内，记忆操作成为与外部决策共同优化的内部行动，使 Agent 能够平衡即时效用与长期一致性和任务性能。

此外，将多模态记忆与结构化世界表示（如空间地图、对象中心图（Singh等人，2023）或技能图（Wang等人，2025c；Feng等人，2025a））集成可以支持跨时间和模态的抽象，同时改善检索效率。最后，记忆和世界模型应在相互强化的循环中共同训练：稳定的、结构化的记忆可以提供长期状态线索以改善世界模型预测，而世界模型可以正则化记忆进化以防止身份漂移、目标不一致和行为不稳定（Savinov等人，2019）。

推进这种协同是构建可扩展、可靠的多模态和具身 Agent 以在复杂现实环境中实现长时程自主性的关键。

9.6 现实世界基准测试与评估

现实世界中记忆启用基础模型 Agent 基准测试的核心挑战在于研究级基准抽象与现实世界部署复杂性之间的持续不匹配，对用户中心和 Agent 中心记忆都是如此。在用户方面，大多数现有基准将长期个性化简化为合成事实回忆，其中 Agent 检索嵌入长上下文或脚本交互历史中的静态用户属性（如角色事实、偏好或对话）。虽然此类设置便于受控评估，但它们未能捕捉真实用户满意度，这取决于数周或数月上的偏好漂移、冲突信号、部分可观察性和延迟反馈。

诸如LoCoMo（Maharana等人，2024）等基准强调长上下文检索准确性，但隐含假设静态用户意图和明确的真值，忽视了如陈旧偏好重用、长期用户状态的不正确覆盖或不安全保留敏感信息等关键故障模式。即使明确针对演化偏好的PersonaMem（Jiang等人，2025a），也在完全模拟会话上评估，并不评估记忆更新或刷新。

在 Agent 中心方面，包括WebArena（Zhou等人，2024）和OSWorld（Xie等人，2024）在内的交互基准通过基于执行的评估改善了现实性，但仍受限于经过筛选设计的环境、重置中心任务设计和短评估时程。最近的扩展开始放松第一个限制：InterruptBench通过任务中添加、修订和撤销用户目标来增强WebArena-Lite，并报告即使对强骨干模型处理它们仍然困难（Zou等人，2026）。

这些约束掩盖了 Agent 是否能跨事件积累、修订和安全利用经验，特别是在非平稳工具、策略或环境下。因此，Agent 可能优化短时程成功，同时在记忆关键能力（如来源跟踪、矛盾解决和长期策略一致性）上无声失败。直接探测隐藏用户状态使这种差距可测量：任务完成率即使对无记忆基线也趋于饱和，而用户演化状态的恢复保持中等并在前 K检索下进一步退化（Ma等人，2026）。

这种差距反映在通用助手基准如GAIA（Mialon等人，2024）中的观察，失败通常不是源于孤立推理错误，而是来自跨多模态和由工具介导的交互中的脆弱状态转换和不正确记忆更新。

未来研究应朝向闭环、纵向和基于执行的评估范式，明确在现实约束下强调持久记忆，对用户和 Agent 都是如此。对于用户中心记忆，基准应纳入具有受控偏好漂移、模糊反馈和真实用户奖励的重复交互，支持直接测量满意度对齐的记忆行为如压缩、选择性遗忘和安全覆盖，而非静态回忆准确性（例如，将用户历史扩展到多个月偏好演化或反事实反馈）。

对于 Agent 中心记忆，评估应超越模拟重置，转向部分开放或持续演化环境，其中经验积累具有真实后果，如金融交易沙盒、长时间运行Web服务或具有延迟回报的竞争控制任务，支持在相同条件下比较记忆增强 Agent 与无记忆基线。基于执行的框架如OSWorld（Xie等人，2024）可以通过记忆敏感不变量扩展，要求 Agent 对持久状态进行版本控制、审计和回滚，并将来源元数据附加到存储知识。

同时，标准化工具调用中介层（如MCP风格接口）可以支持可重现日志记录、权限执行和重放，支持在现实约束下对记忆-策略交互进行细粒度评估。最后，基准应明确量化资源-效用权衡，测量作为 token 预算、存储成本和延迟函数的记忆质量，反映现实部署的有界记忆条件。共同地，这些方向将基准测试从情景任务完成重新框架为对记忆作为核心能力的系统级评估，共同塑造用户信任、Agent 自主性和跨演化环境的长期效用。

10 结论

记忆正成为在长时程、上下文规模爆炸和用户依赖环境（如 Agentic 编程、深度研究和计算机使用）中运作的基础模型 Agent 的关键组件。在本综述中，我们沿三个维度统一了设计，包括记忆载体（内部和外部）、认知机制（感觉、工作、情景、语义和程序性）和记忆主体（用户中心和 Agent 中心），并系统分析了跨单 Agent 和多 Agent 拓扑的操作机制。

我们进一步强调了学习策略在记忆管理中的日益增长作用，显示记忆策略本身正通过提示设计、微调和强化学习被优化，将记忆从被动存储转变为主动进化的能力。

通过评估当前基准和指标，我们识别了记忆效用的关键差距，特别是在现实世界部署中，并概述了六个开放挑战以指导未来研究：持续学习和自我进化 Agent 的记忆、多用户与多 Agent 的记忆组织、记忆基础设施与效率、终身个性化与可信记忆、多模态、具身与世界模型 Agent 的记忆，以及现实世界基准测试与评估。

随着AI进入“下半场”，发展使 Agent 能够从自身经验中持续学习、跨用户和环境个性化、并随时间可靠进化的记忆系统对于在现实世界中实现真正自主和实用的基础模型 Agent 至关重要。