AbsoluteAGI / 研究与报告

持续数字个体研究综述

连续性、自主性与责任

Arthur LEI

正式版 / 2026-11-07

文献检索截止 2026 年 11 月 4 日

摘要

持续数字个体的研究对象，是能够跨交互、任务及运行中断积累经历并持续行动的系统。本综述采用结构化文献映射与批判性叙述综合，检索截至 2026 年 11 月 4 日，阅读了 97 条研究记录和 7 组项目及部署材料，并对读 AI Village、Claude’s Corner、Andon Market 和 The Agentic Scientist 四个案例。

现有研究已能分别观察信息更新与工作接续、给定目标下的主动行动与自选活动，以及推进、核查和纠错在人与 Agent 之间的分工。证据仍主要来自单项基准、短期对照或观察性部署。能读回记录、能发起行动和能持续履行承诺，需要沿工作过程分别验证。研究的关键空白在于，跨版本与跨环境的接续有多可靠，目标取舍在多大程度上受资源和运行条件影响，以及在同等交付要求下，责任委托能否长期减少人的监督与补救劳动。

持续数字个体的研究还应跟踪兴趣、关系和自我认识的成长，使任务表现与个体自身关切的变化能够在同一历史中对读。

目录

摘要 2
第 1 章 引言与研究问题 4
第 2 章 综述方法 5
第 3 章 理论基础与研究演进 8
第 4 章 连续性 12
第 5 章 自主性 17
第 6 章 持续工作与责任委托 22
第 7 章 评估方法与实证发现 27
第 8 章 综合分析与当前挑战 31
第 9 章 未来研究方向 35
第 10 章 结论 38
参考文献 40

第 1 章 引言与研究问题

团队周一把一份研究报告交给 Agent，约定周五发布。周三，修订后的数据使一项关键结论失去依据，最初的会话早已结束。系统会自行发现哪些段落需要重查、继续推进核验，并在交付条件发生变化时请人决定吗？

本文要回答的问题是，一个数字系统如何在跨越多次会话、任务和运行中断后，仍能依据过去的经历作出判断，自行发起行动，并对已经接下的工作持续负责。本文以“持续数字个体”指称这一研究视角：沿着同一系统可辨认的运行历史，考察经历、目标和承诺如何改变后续活动。持续工作是观察这些关系的一个场景，研究对象还包括学习、关系、自选活动与自我表征的变化。具体而言，连续性涉及哪些状态和经历能够延续、修订或转化为可复用的经验。自主性涉及系统如何形成或采纳目标，又在什么条件下选择行动、等待或改变方向。责任委托涉及谁推进工作、核查成果和处理失败，以及谁有权改变目标、接受交付或在必要时接手。

本文采用结构化文献映射与批判性叙述综合。检索截至 2026 年 11 月 4 日，纳入 97 条研究记录和 7 组项目及部署材料，并选取 Claude’s Corner、AI Village、Andon Market 和 The Agentic Scientist 作为四个案例。文献比较以观察单位、时间尺度、目标来源、人工介入和结果测量为共同维度。受控基准用于判断特定条件下的表现，部署调查与遥测用于描述系统的实际用法，案例记录用于追踪跨时间的活动与后果。比较时保留各项研究的任务条件和证据来源。[1] • [2] • [3] • [4]

综合结果显示，系统已能从过去的记录中取用信息、修订部分判断，并在受控条件下接续未完工作。能否在多次中断和版本更换后继续履约，仍需更长的对照。给定目标下何时介入或等待已有可重复的评估，自选活动也已在开放部署中出现，但目标怎样形成、为何持续、受到多少资源与调度条件的影响，仍难从现有记录中分开。推进、核查和部分纠错已可委托给 Agent，现场案例中的标准修订、异常处理和最终确认仍可能需要人参与。持续委托是否减轻了人的负担，要在可比的交付要求下，把工作结果和监督、返工、接手的劳动一起计入。[5] • [6] • [7]

现有文献分别围绕记忆检索、任务完成、开放探索和人机分工积累了大量结果，彼此却经常采用不同的观察单位与时间尺度。本综述据此建立连续性、自主性和责任委托的比较框架，把机制研究与部署材料放回同一段持续活动中理解。其贡献在于连接这些分散的路线，明确结果之间的可比条件，并从交叉处识别纵向研究的缺口。一次回忆正确，说明某条过去的信息可以取回；它能否在约定更新后指导行动，需要观察信息在工作中怎样被使用。同样，一次成功交付和一项长期委托的成效，取决于不同的结果与分工记录。通过把研究结果对应到经历、选择、行动和核验的具体环节，本文解释了为何相近的成绩可能回答不同的问题，也据此指出哪些关系值得在同一个体、同一项委托的纵向研究中检验。

第 2 章说明检索、筛选与比较方法，第 3 章梳理理论基础和研究演进。第 4 至第 6 章依次讨论连续性、自主性与责任委托，第 7 章对读评估方法、实证发现和四个案例。第 8 章综合各路线的发现与证据空白，第 9 章提出相应的研究设计，第 10 章回到开篇的问题，总结已有结论与未解之处。

第 2 章 综述方法

2.1 研究范围与检索路径

本综述采用结构化文献映射与批判性叙述综合。检索围绕第一章提出的问题展开。我们追踪研究者怎样定义和测量经历、状态与承诺对后续行动的影响，怎样识别行动的发起与维持，以及怎样记录推进、核查和纠错在人与 Agent 之间的分配。直接考察持续运行的研究进入这一范围。只考察其中一项机制的工作，只要能解释长期活动中的具体问题，例如过期记忆如何误导后续行动，也纳入比较。认知科学、人工生命与人因研究为概念和方法提供来源，其发现按原研究的对象和环境解释。

首轮检索和前沿补检的截止日为 2026 年 11 月 4 日。经典文献按概念脉络回溯，近期研究重点覆盖 2023 年以来的 LLM Agent。我们使用学术网页检索和面向出版平台的关键词查询发现候选，再到 arXiv、ACL Anthology、OpenReview、ACM、PMLR、NeurIPS、期刊或作者机构页面核对书目与版本。心理学、认知与人因条目同时使用 PubMed 核对书目信息。检索从领域词组和已知题名两端展开，例如把长期记忆与自传记忆、记忆更新相连，把持续工作与状态恢复、人工接管相连。沿已有综述和核心论文的参考文献回溯，再针对关键概念和基准查找后续研究。补检专门覆盖模型更换、在线学习、等待与监控、错误接手，以及公开部署记录。各主题的检索式示例和筛选记录见来源登记。

2.2 筛选、去重与版本核对

筛选时考察材料是否直接回答一个研究问题、能否定位一个概念的来源，或能否提供有明确条件的方法对照。理论论文和已有综述用于梳理概念与研究路线。涉及方法、数据或结果的实证主张，回到原始研究核对。通用排行榜、缺少方法说明的产品宣传和偏离研究问题的搜索结果排除出主目录。任务、计分方式或研究问题不同的材料分别登记，例如 Vending-Bench 的两个版本各有自己的模拟条件和计分规则。

计数以研究作品为单位。同一工作的预印本、会议版和期刊版合并，登记实际阅读版本，以及会改变解释的样本、指标和文本差异。首轮得到 87 条去重记录，前沿补检增加 10 条，形成 97 条出版物及研究报告的阅读语料。另外登记 7 组项目、运行和部署材料，每组可包含多份文件。97 条是阅读语料的规模；文末列出的 85 条是正文直接引用的作品及项目资料。两种计数分别对应阅读范围与引用范围，项目组中的多份报告、日志也可以分别进入参考文献。论文和项目记录若来自同一运行，例如 OpenLife 原论文与后续公开档案，则建立关联。后续材料可以延长观察窗口，同一批运行仍按同一证据来源处理。

我们逐项通读了 97 条研究记录的正文。原文通过正式出版页、作者稿或可读的期刊扫描取得，主要分析所需的表图和附录一并核对。正式版与早期版本同时可读且存在实质差异时，数字随所引版本及其分析样本报告。归档分别保存所取得的 PDF、网页快照和从在线原文提取的文本，索引标记出处和版本。能够核对的范围以台账为准。例如，G12 完整通读了 NBER 工作论文，另在期刊网页核对发表版的样本定义与主要结果表。原文归档索引 • [8]

2.3 证据提取与跨研究比较

逐篇记录研究问题、对象与环境、设计与对照、分析单位、指标和主要结果，以及作者给出的限制。证据定位到具体页码、章节、表图或附录。Claude Code 中配置的 MiniMax 用于分段整理公开原文和定位候选证据。通读与提取分批进行，引用采用前再回到原文核对。台账分别保存辅助提取、直接通读、并行通读后复核的记录。数字冲突保留在证据卡中，核查比较组、分母、估计方法和版本后再决定如何引用。

比较维度	提取内容
观察单位和时间跨度	一次调用、会话、任务、任务序列或同一运行个体。分别记录人类基准任务时长、Agent 实际执行时间和部署观察窗口。
状态及其延续	保存和读取了什么，何时写入、更新或失效。遇到中断、模型更换或环境变化时，哪些记录、状态和未完成事项能够接续。
行动发起与学习	目标的来源、行动的触发条件和可选范围。区分同一任务的再尝试、跨任务复用经验、可执行技能的积累和参数更新。
工作分工与人工介入	谁设定标准、推进、检查、纠错与接受交付。记录主动询问、批准、接管和撤销，以及研究中实际测量的时间或监督投入。
结果与比较条件	记录对照、重复次数、预算、模型和工具配置，以及判分依据。把答案正确、工具执行、真实状态变化、持续一致性和人工负担分别登记。

比较从测量问题开始。例如，原版 LoCoMo 的问答结果采用词元 F1，APEX-MEM 报告的是模型裁判给出的问答正确率。两者可用来讨论记忆系统的设计，分数的直接比较则需要一致的样本和评分规则。同样，相对人类独自工作的提升，与相对人或 AI 中表现较好一方的提升，回答两个不同的协作问题。对照组、结果单位和环境足够接近时进行结果对读，其余材料用于比较机制、适用条件或证据缺口。综述保留原研究报告的效应量及其条件，跨路线以叙述综合为主。[9] • [10] • [11]

2.4 项目材料的纳入与事件编码

项目材料补充受控任务较难呈现的长期过程，例如模型更换后怎样恢复工作、出现运营失误后由谁处理。纳入材料需要能识别项目及来源，并具备可定位的日期、运行阶段、事件、配置或产物。七组材料均按已登记的公开主材料复核，正文重点对读 AI Village、Claude’s Corner、Andon Market 和 The Agentic Scientist。OpenLife 的公开档案补充其论文的观察窗口，Anthropic 的产品遥测与 Project Vend Phase Two 分别提供监督行为和运行条件变化的材料。

正文引用的事件按发生顺序组织，记录可查的触发条件、行为者、行动、结果、后续跟进和人工介入。公开日志或可检查产物、运营方报告、Agent 自述、事后重放分别标明。项目方同时调整模型、工具和操作规则时，时间线用于追踪这些变化及随后的行为，因果判断要看是否有可比较的对照。以 Andon Market 为例，真实门店的经营事件用于分析人与 Agent 的分工，按原情境运行多个模型的重放则用于比较它们在给定条件下的处理方式，两部分分别计数。[3]

2.5 覆盖范围与推断限制

这批语料以公开可访问来源和英文材料为主。检索记录保留了实际查询示例、书目核对、纳入与排除案例，以及采用的原文版本。网页检索结果包含重复出现的记录，现有日志也没有逐数据库导出的完整命中与排除清单。因此，97 条描述的是本次形成并通读的研究语料，无法据此估计整个领域的文献总量或发表比例。较新的预印本、持续更新的项目页面和历史论文的扫描件，各有版本与可读性限制。

证据的解释取决于它如何产生。形式模型支持在其假设下推导的结论，受控基准回答指定环境和评分规则下的表现，现场实验支持其比较设计所能识别的效应，项目日志和遥测则提供实际运行的过程和分布。对项目方尚未开放的完整会话、账本、配置历史与运行轨迹，相关结论限定在公开可核对的层级。AI Village 的受限数据集、Claude’s Corner 的底层会话分析数据，以及 Andon 的完整经营账本均属于这一情况。后续章节据此区分哪些结论已有多种证据支持，哪些仍取决于单一环境、单个部署或尚待检验的机制解释。

第 3 章 理论基础与研究演进

今天的语言 Agent 可以根据一次工具调用的结果改变下一步行动。把观察窗口拉长，问题就变了。它可能在几天前接下任务，中途形成了新的判断，还要在环境变化后辨认自己原来的承诺是否有效。要解释这一段行为，既要看系统如何作出当下的决定，也要看是什么让不同时间的决定彼此关联。早期 Agent 研究从意图与承诺入手，认知架构从记忆与行动控制入手，人工生命从个体如何维持自身入手。它们各有起点，后来才在同一类系统上相遇。

3.1 经典 Agent、意图与承诺

20 世纪 90 年代的 Agent 研究已经提出，系统在变化的环境中既要响应事件，也要有能力主动追求目标。Wooldridge 和 Jennings 在 1995 年的综述中，将自主、社会交互、反应和主动性列为 Agent 的一组基本属性。这里的自主，首先指系统对自己的行动和内部状态具有一定控制。反应式架构可以迅速处理环境变化，审议式架构则可以依据对世界的表征推演后续行动。混合架构把两者放在一起后，还需要决定何时让突发事件打断原计划，何时继续执行。 [12]

Cohen 和 Levesque 在 1990 年从时间上解释了意图的作用。一项选择一旦成为意图，就会约束后续的选择和尝试。执行失败后，如果行动仍被认为可行，原意图可以继续有效。反过来，如果系统相信目标已经实现、已经无法实现，或支撑承诺的条件已经消失，就有理由解除承诺。关键在于保留**判断承诺是否仍有效的依据**。例如，系统把采购请求发出后，仍需根据送达或交付记录判断目标有没有实现。形式模型刻画的是信念与承诺之间的关系，现实中的履约还取决于信念能否被可靠地更新。 [13]

Rao 和 Georgeff 将这类问题推进到工程架构。BDI 将信念、欲望与意图区分开，分别表示系统对情境的认识、可能追求的目标和已经选定的行动方向。他们的实现讨论使用预先编写的计划及触发条件，并允许多个意图在运行中继续执行或挂起。这种取舍在实时系统里尤其具体。每次收到新信息都从头规划，可能让工作无法推进。一直沿用旧计划，又可能错过使原决定失效的事件。这条路线给持续工作留下了一个重要区分。提出候选目标、承诺实现目标、在失败后继续尝试，是不同环节。目标为什么会进入候选范围，则引向动机研究。 [14]

3.2 认知架构与行动循环

与意图逻辑并行，认知架构研究问的是，系统内部怎样把知识组织成连续的行动。1987 年的 Soar 将问题求解组织为一系列操作选择。已有规则无法判断下一步时，系统把这个僵局转成一个子目标，例如先解决两种操作该选哪一种。解决子目标后，Soar 能沿推理依赖提取真正起作用的条件，把这次解法写成以后可直接应用的规则。行动遇到的困难会改变接下来的内部活动，求解经历又改变以后行动的方式。循环中可以自动产生子目标，其范围取决于当前任务和系统已有的知识。 [15]

ACT-R 从另一端研究整合问题。感知、动作、目标和陈述性记忆等模块可以各自处理信息，再通过缓冲区参与规则选择。在 Anderson 等人的 2004 年论文中，模块安排具体到可以与人的反应时和练习过程对照。例如，两项任务使用不同的感知和动作通道时，部分加工可以并行。需要争用同一资源时，干扰便会出现。与 Soar 的求解和规则学习相比，ACT-R 还要求架构解释不同心理过程如何在时间上配合，并让这套解释接受行为数据的检验。 [16]

认知架构里的记忆首先服务于当前决定。关于经历怎样塑造此后生活，心理学又提供了两条互补路线。Conway 和 Pleydell-Pearce 的自我记忆系统把回忆理解为一次重构。当前目标影响哪些线索会被追寻，已有的自传知识反过来约束哪些目标和自我理解能成立。这使记忆的作用超出了提供事实。记得某次合作经历，可能改变以后愿意接受什么任务。眼下关心的事也会改变从那段经历中想起什么。这个双向关系为考察数字个体的经历如何进入后续目标提供了理论参照。[17]

互补学习系统理论则从学习速度与干扰出发。McClelland 等人用模型说明，快速保存单次经验和缓慢提取跨经验的规律，可以解决不同问题。如果在已有知识上集中学习一个新例子，旧知识可能受到干扰。把新旧材料交错呈现，则有机会保留原有结构。随后关于睡眠与巩固的研究进一步区分了减少干扰、选择性再激活和系统层面的重组。由此可以提出一个工程问题。一段经历保留下来之后，系统何时需要重新处理它，处理结果又能否影响后来遇到的任务？ [18] • [19]

3.3 内在动机、自生产与适应

发展机器人与强化学习把目标的来源变成了可以实验的问题。Oudeyer 和 Kaplan 区分了按知识变化和按能力进步组织探索的机制。前者可以让新颖或难以预测的情境更值得尝试，后者关心系统在某类目标上的能力是否正在进步。两种机制会把系统引向不同地方。持续产生随机变化的对象可以很难预测，却未必提供可学习的结构。后来的好奇心实验据此让预测误差的计算避开部分与行动无关的输入，但在局部获得新颖性信号，仍可能不足以引导跨越多个阶段的探索。 [20] • [21] • [22]

Colas 等人把研究单位进一步推进到可选择的目标。系统可以产生或选择练习目标，尝试实现它们，再根据能力变化安排下一轮课程。目标的表征、可选范围和成功判据仍可由设计者提供。自选目标有不同层次。一些设计让系统自行决定下一步练什么，另一些还让系统学习如何表征或评估目标。基础模型让目标能够用自然语言或代码表达，候选目标因而可以组合得更复杂。这扩大了可研究的范围。如何形成长期方向、在不同目标之间取舍，以及目标能否随着经历改变，仍需要纵向观察。 [23] • [24]

还有一类研究从系统自身的状态出发。Keramati 和 Gutkin 的稳态强化学习把行动的价值与内部状态偏离设定水平的程度相连。同一种结果在不同的内部状态下可以具有不同价值。Marsella 和 Gratch 的 EMA 模型则从情境评价出发。事件怎样影响既有目标、是否还能改变结果，会随系统对情境的理解而改变，并影响随后采取的应对方式。一个关注内部需要，另一个关注事件对目标的意义。两者都允许行动结果改变下一轮行动的条件。 [25] • [26]

人工生命研究追问的是更早一步的问题。为什么某些状态对这个系统有意义？Maturana 和 Varela 用自生产（autopoiesis）描述一种组织。系统中的过程不断产生使这些过程得以继续的成分，并由此维持个体边界。个体的具体结构可以改变，维持它作为一个个体的组织关系则延续。Di Paolo 在此基础上强调适应性。除了经受扰动，系统还可能对趋近生存边界的趋势作出反应，改变与环境交换的方式。Barandiaran 等人进一步把个体如何成立、何种结果对它有利，以及它如何主动改变互动条件，放进同一个行动者理论。由此，问题从「什么信号引导探索」扩展到「这种行动如何服务于一个个体得以继续存在的条件」。 [27] • [28] • [29]

资源实验把这条理论路线带到可观察的环境。Masumori 和 Ikegami 在 Sugarscape 模拟中设置资源、消耗和退出条件，比较 Agent 在不同提示和条件下如何行动。后续 OpenLife 直接引用自生产与适应研究，并把预算与活动放进开放网络中的实际运行。两种环境改变了证据的性质。

模拟允许研究者操纵资源条件。部署可以观察 Agent 在既有基础设施、人工维护和预算供给下如何改变活动。OpenLife 报告了外部收入，也记录了持续运行所需的外部补贴。收入、成本与补贴放在一起，才看得出个体的行动在多大程度上维持了它的运行。 [30] • [31]

3.4 从工具调用到具备记忆和状态的 LLM Agent

ReAct 的关键变化发生在单次任务内部。模型可以先形成一段中间推理，发出环境动作，再根据观察修正后续判断。比如，检索结果可以纠正原先的判断，也可能让系统陷入反复搜索。任务中的决策由此开始受执行过程约束。Sumers 等人的 CoALA 随后明确把这类 Agent 放回认知架构的历史。它区分工作记忆与长期记忆、作用于环境的动作与作用于记忆的内部操作，也允许系统在执行前反复生成和评价候选行动。ReAct 实现了其中的一部分。CoALA 提供了一种比较方式，让研究者能指出不同系统在哪些环节作出了设计选择。 [32] • [33]

记忆的加入把时间范围从一次调用延伸到会话和任务之间。Generative Agents 把观察、反思与计划写入记忆流，在模拟社会中让过去的交往影响后来采取的行动。MemGPT 处理的是上下文容量有限的问题。系统需要把信息在活跃上下文和外部存储之间调度。前者着重选择什么经历值得带入下一次行动，后者着重长历史如何在有限容量下可取用。接下来的 LoCoMo 与 LongMemEval 把能力拆开检验，例如信息能否跨会话组合，更新后是否还沿用旧事实，证据缺失时能否恰当地不作回答。这使研究可以追问记忆在哪一环失效。 [34] • [35] • [9] • [5]

从记忆到学习的转折，在于经历被保留成什么，以及它在何时能派上用场。Reflexion 把一次失败的反馈整理成下一次尝试可读的反思。Voyager 在给定的探索方向下选择具体任务，把成功行为保存为可调用的代码技能。Agent Workflow Memory 从操作轨迹归纳可复用的步骤，Evo-Memory 则在有顺序的任务流中考察过去的经验能否帮助之后的任务。读回一段历史、复用一套技能和在不同任务中持续改进，需要不同的比较。特别是第二次尝试带来的进步，只有在反馈、尝试机会和可用经验等条件厘清后，才有助于判断跨任务学习。 [36] • [37] • [38] • [39]

经历的影响也开始通过不同方式接受检验。ID-RAG 将身份信息和具体事件分开存取，在模拟中考察行动是否符合预设的身份档案。它让自我表征成为可操作的设计变量，也留下了档案怎样随经历更新的问题。ASCENT 则从另一端推进，把经验证的执行经历转为保留到后续任务的参数更新，并在每次更新前先计分。前者考察既有自我表征如何进入决定，后者考察既有行为如何改变后续能力。两条路线的连接点在于，系统能否在学习新的做法时，仍辨认和维持先前的经历、目标与承诺。 [40] • [41]

执行研究也逐渐细化了什么算完成。SWE-bench 用仓库里的测试检验补丁能否修复问题并保留已覆盖的原有行为。PaperBench 把写出实现、在新环境中运行、复现论文结果分开计分。这种拆分把经典意图研究留下的问题变成可观测的工程问题。一项行动结束后，系统需要依据什么判定任务已经完成？当检验发现偏差时，后续还需要决定是继续修正、调整计划，还是将问题交回给人。 [42] • [43]

持续工作还带来了运行层的问题。早期的 Sagas 研究已经指出，长事务分步执行后，失败时可能需要用有业务含义的补偿操作处理已发生的后果。今天的 Agent 恢复研究同样需要同时处理多种状态。AgentRewind 在受控环境中把 Agent 的上下文和工作区一起恢复，近期针对 LangGraph 项目的研究则检查普通重启与容器替换后，运行中的工作能否接续。两条路线可以相

互参照。恢复上下文让系统知道原来做到哪里。恢复或补偿外部状态，决定原来的操作还能不能安全地继续。[44] • [45] • [46]

3.5 路线汇合与概念分歧

第一处汇合发生在**目标与经历**之间。意图研究提供了维持和解除承诺的条件，认知架构提供了在执行中读写状态的机制，动机研究则解释候选目标怎样获得优先级。把三者放在一起，便能研究系统因何选择去做一件事、执行中怎样依据新信息调整做法，以及结果怎样影响下一次判断。相似的外部行为可以来自不同机制。系统自行发出提醒，可能是在响应预设事件，也可能是在追踪先前承诺。长期观察需要保留目标和触发条件，才能区分这些过程。

与此相邻的是**何时介入**的问题。Horvitz 的混合主动研究从用户意图的不确定性、行动收益与打扰成本出发，考察系统何时行动、何时询问和何时等待。后来，ProEvent 把事件状态随时间变化纳入评估，检查提醒是否在合适的时点发生，以及系统是否会在无需介入时误触发。这条路线为自主行动补充了时机和交互成本的尺度。系统可以准确判断何时值得提醒，同时沿用用户设定的目标。它的主动性体现在如何选择介入，目标从何而来则由另一组证据回答。[47] • [48]

另一处汇合发生在**工作自主与人的控制**之间。Bainbridge 早已指出，自动化接管日常操作后，人可能仍要监控系统并在最难处理的异常中接手。Parasuraman 等人把信息获取、分析、决策和执行分开，说明这些环节可以具有不同的自动化程度。对持续工作的 Agent，也可以分别问谁发现异常、谁判断风险、谁有权继续执行，以及谁接受成果。[49] • [50]

随着系统能从经验中学习，Matthias 将其中一项困难表述为责任缺口。设计和部署系统的人可能难以预见它学习后的具体行为，传统归责所依赖的控制因而变得难以确定。之后，有意义的人类控制框架提出两个可以分别考察的条件。系统能否对相关的人类理由与环境事实作出反应？能否追溯到理解其能力、后果及自身责任的人？在 Agent 的长期工作中，日志可以帮助定位某次失败。要解释谁有条件改变决定并对后果负责，还需检查当时的分工、信息和权限。[51] • [52]

分歧最深的地方在于，**何种证据能够支持对个体的何种判断**。记忆研究可以检查先前经历是否正确进入后续行动，人工生命可以追问个体是否参与维持自身的存在条件。关于主体经验，Butlin 等人从意识科学的不同理论推导出候选指标，例如某种形式的循环处理、信息广播，以及系统如何利用行动与感知之间的关系。指标的解释取决于所采用的理论、系统边界和机制证据。由此，第一人称自述、跨时行为延续、自我维持与意识指标各有自己的证据来源。一个系统长期保持连贯的自述，首先提供的是关于自我表征与表达的观察。[53] • [54]

这些路线的汇合，使持续数字个体可以作为一个跨时间的研究对象。沿着同一条运行历史，可以追问某段经历怎样留下影响，目标为何在后来被维持或改写，行动如何改变继续运行的条件，以及谁在结果出现后负责核查和纠正。

第 4 章 连续性

设想一个 Agent 和同事约定周五发布一份报告。第二天，他们发现数据还需要复核，于是改为周五内部审阅，公开日期待定。当 Agent 再次参与这项工作时，复述最初的约定只能说明它找到了过去的信息。它还要知道哪份约定当前有效、修改是因何发生的、复核做到哪一步，以及自己曾向谁承诺交付什么。如果它后来遇到另一份报告，能在排期前先确认数据是否齐备，这段经历就进一步改变了它的做事方法。

这一例子涉及三种可以分别观察的延续。过去的经历能否参与当下的判断，属于记忆问题。个体能否把经历与自己的角色、目标和承诺联系起来，属于自我表征问题。执行环境变化后，已经完成的工作能否保留，余下事项能否继续推进，属于运行问题。三者在长期运行中相互依赖，也可能各自失效。一份完整的日志可以随备份迁移，迁移后的系统仍可能按旧约定发布报告。理解这种分离，是比较本章各类研究结果的起点。

4.1 经历、记忆与自我表征

经历首先以可追溯的事件进入系统，例如一封邮件、一次工具调用、一条收到的反馈。记忆则把其中一部分组织成日后可以使用的内容。同一次延期可以留下多种记忆。事件记录保留对话和先后次序，事项记录更新当前交付约定，经验记录则提炼出“承诺公开日期前先确认复核条件”。这些记录各有用途，也各有失真的可能。摘要若只写“周五交付”，就抹去了公开发布与内部审阅的区别。经验若进一步写成“这位同事总会推迟工作”，又把一次有具体原因的延期变成了对人的笼统判断。

Conway 与 Pleydell-Pearce 的自我记忆系统模型为理解这种选择提供了一个起点。在这个模型中，当前目标参与组织自传体记忆的提取，过往经历也为个体理解自身目标提供材料。负责协调进度时，延期经历首先提示哪些事项需要重排。后来回顾自己的判断方式时，同一件事可能提醒个体，自己曾低估上游证据的准备时间。McAdams 与 McLean 对叙事身份的研究把视野延伸到个体如何把重新理解的过去与设想中的未来连在一起。两条路线共同提出了一个对 Agent 研究有用的问题。系统在不同情境中可以重新解释同一经历，它能否同时保留依据，让后来的解释还能回到原始事件核对？ [17] • [55]

在 Agent 系统中，这种关系可以通过不同机制进入行动。Generative Agents 把观察、计划和反思纳入记忆流，并以相关性、近期性和重要性选择当前可用的信息。反思可以把多次观察组织为一条更抽象的理解，并关联支撑它的记录。ID-RAG 则把人物的立场与关系预先组织成图，让角色在模拟竞选活动中检索与当前决策有关的内容。两者的比较揭示了自我表征的两个方向。一个方向是从经历中归纳新的自我认识，另一个方向是让已经形成的角色与价值取向参与行动。前者容易把有限观察概括过头，后者则需要面对既有自我认识何时应当修订的问题。 [34] • [40]

这里所说的自我表征，是系统用来关联自身经历、当前角色、目标和承诺的可检查信息。它是否有效，取决于这种关联能否在后续行动中产生差异。例如，系统既能说明“我答应做数据复核”，也能在工作重排时把这项义务继续放在自己的待办里。Klein 与 Nichols 讨论的神经心理个案提供了一个概念上的区分。个案能够提取属于自己过去的事件信息，却报告缺少以往回忆时那

种经历属于自己的感受。对数字系统的研究目前能够追踪记录、归属判断和行为之间的联系。第一人称日志在其中可以作为自我表征的材料，其证据价值还取决于系统随后怎样使用它。 [56]

4.2 跨会话记忆、过期信息与修订

跨会话记忆最早遇到的困难是，系统在新的交互里如何找到足以回答问题的旧信息。MemGPT 将可直接提供给模型的上下文与外部存储区分开来，允许系统按需调入记录。这个机制解决了记录如何在有限上下文之外保存和调用的问题，也把另一项工作留给了检索与推理。报告约定的修改可能分布在两次会话和一封邮件中。只找到第一次会话，系统会得到一个真实的旧约定，也可能因此作出错误的当前判断。 [35]

LoCoMo 用多会话对话考察跨时间的信息整合。LongMemEval 进一步把跨会话推理、时间推理、知识更新和证据不足时的弃答分开评分，使错在何处更容易辨认。EverMemBench 则把说话者归属和多人对话交错纳入评估。在其表 4 和表 5 中，Gemini 3 Flash 的多跳回忆得分在全上下文条件下为 26.51%，给定标准证据后为 88.37%。这个同模型对照显示，在该题组里，从混杂记录中定位并组合证据本身就构成了显著难点。对实际系统而言，排查时可以先看支持结论的记录是否进入了当前上下文，再看模型是否正确使用它们。 [9] • [5] • [57]

当新旧记录都能找到时，系统还需要判断它们的有效性。遗忘与修订是两种不同的操作。MemoryBank 用随时间衰减和随回忆强化的机制调整记录的可提取程度。这有助于管理记忆的显著性，但一条经常被引用的安排也可能在一次正式修改后立即失效。Memora 把这种失效纳入评估，用遗忘感知记忆准确率（FAMA）惩罚继续依赖过期或已被否定的信息。APEX-MEM 则采用追加事件的方式保存变化，在检索时利用时间关系解释记录。在报告的例子里，系统回答“当初怎么约的”时应找到旧版本，决定“周五该做什么”时应使用当前版本。旧约定仍能解释计划为什么改变，它对行动的约束已经结束。 [58] • [59] • [10]

时间顺序也只提供了部分依据。较晚的记录可能是建议，也可能是有权修改安排的人作出的确认。执行前，系统还要分辨信息来源、变更权限，以及哪些事实能从外部核对。StateWise 在受控条件下扰动单条持久记录，并分别观察系统能否自行核验事实、在涉及用户意图时请求澄清，或在无法可靠修复时停下。这把研究推进到了一个更接近实际工作的环节。记忆里的冲突可以在问答中体现为答错，到了执行环节，则可能变成向错误对象发送材料，或完成一项已经取消的工作。 [60]

4.3 从经历到学习

经历会以不同方式影响下一次行动。最直接的一种是找回一件相似的往事。Reflexion 把对先前尝试的语言反馈带入后续尝试，帮助 Agent 在重试时调整做法。另一种方式是把多次经历整理成可以反复调用的方法。ExpeL 从执行轨迹提炼经验，Voyager 则将一部分在环境中获得的能力保存为可执行技能。它们保留了不同粒度的内容。一次失败记录说明某个方案在特定条件下出了什么问题，经验规则需要概括适用条件，技能代码还需要在调用时满足环境和输入要求。材料越容易复用，越值得检查其中有哪些条件被省略了。 [36] • [61] • [37]

要衡量这种积累，关键是让后续任务有机会用到此前获得的经验，同时能分辨究竟继承了什么。Evo-Memory 将任务排列为连续到来的序列，比较记忆如何在运行中积累并影响后续解题。Continual Learning Exam 把专业任务组织成逐级递进的工作，后续阶段可以利用先前的产物和知识。这样的设计反映了实际工作中任务彼此关联的特点，也带来一个解释上的难题。后一次表

现改善，可能因为工作区里已经有上一阶段的成果，也可能因为 Agent 学会了一个可以换场景使用的方法。论文提出，若要进一步估计经验的贡献，可以在匹配的条件下重置系统积累的知识，再作比较。对报告例子而言，继续使用已经整理好的数据，和拿到另一份数据时主动补上复核 环节，分别体现了产物的继承与方法的迁移。 [39] • [62]

经验还可以进入模型参数。ASCENT 在任务首次计分之后，用经验证并符合条件的轨迹进行在线更新，保留低秩适配参数供后续任务使用。先计分再更新，使该任务的成绩与此后利用它学习的效果可以分开考察。这条路线与外部记忆的差别，在更换模型时尤其明显。文本经验和技能文件可供继任模型读取或调用，随模型积累的适配参数则依赖兼容的模型基础。前者容易追查某条经验来自哪里，但能否用对取决于检索和应用。后者可以改变后续的行为倾向，某次具体变化来自哪条经验则更难单独追溯。 [41]

连续学习还要求系统处理相互冲突的经验。互补学习系统理论区分快速获得具体经历与缓慢整合规律，睡眠与记忆研究则考察离线阶段对记忆巩固的作用。对 Agent 设计而言，这些研究启发了把日常记录与周期性整理分开的方案。整理时可以保留支持某条规则的经历，也可以把后来发现的反例补进规则的适用条件。以报告延期为例，如果 Agent 后来发现有些材料可以先发摘要，原来的经验就可以从“一律等待全部复核”改为“先核实哪部分支撑当前要公开的结论”。判断整理是否有效，应同时观察新经验能否改变后续做法，以及原先已经可靠的方法是否在整理后仍然可用。 [18] • [19]

4.4 运行状态、中断与模型更换

运行中断使一个平时容易忽略的问题变得具体。Agent 对工作进度的记录，与环境中已经发生的变化，可能停在不同的位置。假设 Agent 调用邮件服务发送审阅稿，服务已经接受请求，进程却在写入发送结果前退出。重新启动后，本地仍显示“待发送”。如果它只按本地状态继续，就可能发出第二封邮件；如果它把一次调用当成成功，也可能漏发。能否接续，取决于系统能否保留调用标识、查询服务端实际结果，并据此判断剩余动作。身份和交付约定解释为什么要发这封邮件，检查点和外部回执决定这一步现在还要不要做。

《Built to Remember, Shipped to Forget》从部署配置、故障注入和开发者修复记录三个角度研究了这类问题。论文在 12 个选定系统上诱发部署事件，16 项预测会违反运行约定的检查中，有 14 项在运行时表现为故障。论文也分别检查了进程重启与容器替换对持久状态的影响。能在原容器中重启后读回的记录，在替换容器时仍可能随本地存储一起丢失。这个差别说明，恢复表现必须连同具体的故障类型、存储位置和外部依赖一起描述。 [46]

恢复研究还区分了从原处继续和退回到一个已知可用的位置。AgentRewind 对齐保存 Agent 的上下文与受控工作环境的状态，并将失败尝试留下的教训带入新的执行分支。在 50 个成对比较的失败终点中，从相同上下文和环境出发，直接继续的恢复率为 8%，使用 AgentRewind 为 30%。这种方法尤其适合研究可受控恢复的工作区。邮件发送、远程审批或第三方系统写入一旦生效，即使本地工作区已经回退，外部结果仍然存在。数据库研究中的 Sagas 提供了另一个思路。长事务分步提交，若整体执行中止，则通过事先定义的补偿操作处理已经提交的步骤。对跨系统工作的 Agent 而言，恢复可能需要查证结果、完成余下工作，或按业务规则补救已经发生的动作。 [45] • [44]

模型更换在状态恢复之外又引入了一项变化。即使日志、待办和工具权限完全保留，新模型也可能以不同方式提取重点、解释旧承诺、判断工具调用是否成功。比较时可以从同一份已知状态启

动，先观察原模型重启后的表现，再看换模型时哪些行为发生改变。对于报告协作，值得追踪的是新模型能否认出当前有效的交付约定、核实发送状态、继续完成复核。若它换了一种语气写日志却正确完成了交接，文体和工作接续就给出了不同的结果。反过来，写得像原来的个体，仍可能掩盖它对旧约定的误解。

4.5 部署个案 Claude’s Corner

Claude’s Corner 提供了一组可以具体比较这些差异的公开记录。部署通过定时调度开启会话，Agent 读取工作区日志并制作可发布的网页等产物。项目报告描述了 60 次会话，其中第 54 至 60 次生产会话作为同期对照。移除日志的条件仍允许读取过去的产物。研究者也比较了任务预先指定、以及允许保留日志但禁止写反思的条件。这些安排使同一个部署中不同载体承接的内容可以分别观察。 [1] • [63]

四种条件下，新网页都保留了原有的配色。现成网页提供了可以模仿的样式，运行提示也要求遵循已有页面的设计。生产对照则出现了另一类行为。第 56 次会话注意到自己连续做了多项数学可视化工作，第 57 次主动转向计算人文。无日志条件下，Agent 也能通过盘点旧产物来避免重复，但该条件下的七次会话仍都落在数学和复杂性相关主题。这个短窗口里的观察表明，维持网站样式、避开已经完成的题目，以及根据对自己近期轨迹的理解重新选题，可以通过不同的信息来源实现。禁止反思的条件只有六次完成会话，日志、反思和自主选题的作用也有待在更多干预组合下分开检验。 [63]

随后发布的文体研究比较了 Opus 4.5 时期的 72 篇日志，以及两个模型在同题下有无旧日志样本时生成的文本。提供旧日志样本后，两个模型生成的文本都更接近原有日志的文体。实际更换为 Fable 5 后，第 73 次会话在能够访问原工作区的情况下写下首篇新日志。按文章使用的文体距离，数值越小表示越接近。这篇日志距旧日志语料为 1.13，距没有旧日志提示的 Fable 5 文本为 0.94。这个单次结果显示，文件可读与文体保持可以分离。由于这次会话本身就在研究文体变化，日志格式也会影响指标，还需要更多普通工作会话来判断差异是否持续。若要进一步判断工作是否接续，证据还须来自换模前后同一批待办和承诺的实际履行记录。 [64]

4.6 当前证据与局限

现有文献提供了几种互补的观察方法。记忆基准适合定位证据丢失、提取失败和时间判断错误。序列任务可以观察经验是否改变后续做法。故障注入可以检查工作在中断前后是否保持一致。部署日志则允许追踪一个系统怎样在多次运行中调用过去，但往往同时受提示、工作区产物、工具权限和模型变化影响。把几类结果放在一起时，首先要明确每种方法真正观察到了哪一种延续。

研究对象	已有方法能观察什么	仍需区分什么
经历与记忆	跨会话找回事件，把修改前后的信息用于相应问题	找到正确记录，与按其有效性采取正确行动
经验与学习	在后续任务中重用反馈、技能或参数更新所得的能力	延用已有产物、熟悉同类题目，与把方法迁移到新任务
自我表征	角色和过去经历是否影响行动，系统是否依据经历调整对自身的描述	旧文本和产物可供模仿，与经历确实改变目标、承诺或选择
中断与接续	恢复上下文和工作环境后，能否继续完成任务	本地恢复成功、外部效果得到核对，以及未完成事项真正接续
模型更换	保留工作区后，比较新旧模型的文体、判断和行动	同一份记录仍然可读，与新模型持续履行原有承诺

文献中相对少见的是，把这些变化放进同一段长期经历里观察。例如，让同一项跨会话工作先经历一次约定修订，再经历进程中断，最后在更换模型后继续推进。研究者可以追踪每个阶段哪些记录仍被调用，哪些行为保留了过去的影响，以及错误最早发生在哪一步。若有匹配的对照，还能进一步区分，起作用的是旧日志、当前事项记录、工作环境的产物，还是在先前工作中获得的做事方法。这类研究有助于解释系统在变化中靠什么保持连续，也能定位为什么某种连续性在一次看似普通的更新后消失。

连续性由此与下一章的自主性问题相接。过去的经历能够进入当下判断，未完成的事项也有机会跨过中断被接着处理。接下来要考察的是，在没有新的任务提示时，什么会促使系统重新想起一件事，并在多个仍然重要的事项之间决定先做哪一件。

第 5 章 自主性

第四章讨论了一个 Agent 如何在重启后接着处理一份待审报告。再往前推一步。距离约定的审阅时间还有两天，数据一直没到，也没有人发来新的指令。Agent 可以维持等待，可以先整理已有证据，也可以发现复核工作反复卡在同一个环节，决定试做一份可复用的数据检查表。这三种选择分别涉及何时行动、怎样推进已有目标，以及是否形成一个值得继续追求的新目标。

本章沿着这三个问题考察自主性。运行调度决定系统什么时候有机会作判断，目标形成机制决定什么会成为它接下来想解决的问题，行动反馈则决定它会继续投入、改变做法，还是放下这件事。已有研究分别把资源状态、学习进展、环境变化和社会互动引入这一过程。比较这些研究，需要追溯一项行动由什么触发、当时有哪几种可选行为，以及观察结果能否解释系统为什么选择了其中一种。

5.1 内部需要与目标形成

目标的来源和目标得到维持的原因，可以分别考察。研究者可以规定系统每次调用都要消耗预算，让系统自行发现一种节约成本的方法。此时维持预算的条件由设计者设置，具体要改进的工作则由系统提出。系统也可以收到一项明确任务，随后因为多次推进受阻而自行调整优先级。Molinaro 等关于自生成目标的综述将目标的来源与动机依据分开，强调目标的表示、生成、选择、追求和修订都可以成为研究对象。这样便能区分系统提出了多少想法、实际选了哪些，以及什么使一个想法经过多次运行仍然受到重视。 [24]

维持自身状态是形成目标的一条路径。Keramati 与 Gutkin 的稳态强化学习模型把内部状态与适宜状态之间的距离写进奖励。一次行动如果能缩小偏差，就会在模型中获得价值。模型还允许预期中的状态变化影响当前行为，因此系统可以在资源实际耗尽前开始准备。在数字系统里，剩余算力、余额或运行时间都可以作为可观测变量。只有当这些量的变化进入后续取舍，才有行为证据表明资源状态参与了目标选择。例如，同样接到两项可选工作，余额充足时系统尝试收益不明的研究，余额下降后它先做能更快结算的工作。观察到这一差异后，还需辨认它来自模型对资源的判断，还是来自一个直接切换任务的阈值规则。 [25]

内部状态也可以表达系统对事件意义的评价。Marsella 与 Gratch 的 EMA 模型考察事件与既有目标的关系，以及系统怎样在重新判断局面时改变应对方式。一项依赖延迟，如果仍能通过改排顺序来处理，可以促使系统寻找替代方案；如果系统判断当前条件下无法补救，可能转而等待、求助或调整计划。这类评价机制的研究价值，在于它把事件如何改变系统对局面的理解，与这种改变如何影响下一步行动联系起来。对持续数字个体而言，同一个结果还可以在以后重新起作用。例如，反复经历相似的交接延误，可能使它把改善交接流程提升为一项长期工作。 [26]

因此，一个目标是否由个体形成，可以沿着一段时间来观察。它在什么情境下被提出，为什么压过当时的其他候选，后来遇到何种证据而被保留、修改或搁置。检查表的例子里，第一次提出设想是一个事件。如果它在下一次发现同类延误时主动翻出设计、试用并根据结果修改，研究者就有了更强的依据，将这些行动理解为一项持续的目标。自然语言扩大了目标能表达的范围，也使这种跨时间的证据更重要。系统说“以后要改进交接”，和它在几次具体取舍中为这件事保留时间，提供的是不同材料。 [24]

5.2 开放探索及行动反馈

另一些目标来自探索。Oudeyer 与 Kaplan 区分了多种计算上的内在动机，将新奇性、预测误差和学习进展等信号放在一起比较。这几种信号会引出不同的行为。最陌生的地方可能吸引一个追求新奇的系统；最能改善当前能力的地方，则可能是已经略懂、还有余地继续学习的地方。这个区别在真实运行中很具体。Agent 可以花一下午浏览不断更新的资讯，也可以用这段时间尝试一种自己尚不熟悉、却有望解决反复出现的数据问题的方法。前一种活动不断提供新输入，后一种活动可以用随后任务中的表现来检验收获。 [20]

好奇心驱动的强化学习把这种差异做成了可以比较的实验。Pathak 等使用对动作后果的预测误差鼓励探索，并在学习出的特征空间里预测，以减轻无关视觉变化的影响。Burda 等的随机网络蒸馏使用固定目标网络，为不熟悉的观察提供探索奖励，并讨论了前向预测容易受到随机变化吸引的问题。一个经典例子是不断播放随机噪声的电视。它持续提供难以预测的画面，系统却未必学到了可以用于其他行动的规律。由此看，评估开放探索至少需要把访问新状态、获得新能力和把能力用到后续任务分开。只统计活动量，会把有成果的探索和对不确定输入的反复追逐混在一起。 [21] • [22]

自设目标的强化学习把问题推进了一步。Colas 等讨论的系统能够从可表示的目标中生成或采样候选，根据自身进展选择练习对象，并使用执行结果更新后续选择。设计者提供的目标空间、反馈和环境仍然决定了实验中哪些目标可以出现，系统则可以在这个范围里形成自己的练习次序。基础模型让候选目标能够用自然语言、代码等方式表达，还可以提出新的任务和判定办法。相应的困难是，目标描述越开放，越需要检查判定标准与实际结果是否一致。检查表写完了、检查表在一次使用中发现了问题、检查表减少了随后多次协作的返工，是三个不同层次的结果。 [23] • [24]

本综述据此区分反馈可能产生的三种变化。方法调整发生在目标仍然有效时，例如更换一种数据核对方法。优先级调整发生在目标仍然可取、但此刻有更紧迫的事情时。目标修订则改变了希望取得的结果，例如几次试用发现问题主要出在版本交接，系统因此放下通用检查表，改做一套交付记录。评估时如果只在末尾给出一个任务成功分数，这三种变化就很难辨认。目标版本、变更前能看到的证据和变更后的行动，能够帮助判断系统是在学习、重新安排工作，还是偏离原先的目的。

5.3 外部事件、主动介入与等待

主动行动经常由外部事件触发。邮件到达、协作者完成一项依赖、监测值越过阈值，都可能给系统提供新的行动理由。在这些情境中，目标可以早已由人交代，何时推进则交由系统判断。定时唤醒、事件订阅和周期检查给判断创造机会。研究上更有辨识力的问题是，在获得同样的运行机会时，系统能否根据当时的证据选择立即处理、继续观察，或保持沉默。Horvitz 对混合主动式界面的研究已把预计的帮助价值、判断的不确定性和打断人的代价放进同一个决策问题。 [47]

SentinelBench 把等待和反应放进可重复的监测任务。它包含 10 个随时间变化的合成网页环境和 100 项任务，其中 20 项直到模拟结束也不会出现目标事件。对这 20 项任务，计分依据是系统有没有误开用于通知用户的表单。被动任务可以在条件成立时通知，主动任务还需要打开页面、核实状态或执行操作。论文比较任务完成率、反应时间和资源消耗，并对照间隔轮询与按条件等待等方式。在表 4 的一个配置中，GPT-5.4 使用休眠工具，在无事件、被动监测和主动处

理任务上的成功率分别为 70%、76% 和 60%。同一个系统在等待、通知和实际处理上的表现可以不同，分别计量有助于定位问题出在哪一步。[6]

面向人的主动性还多了一层。帮助内容确实相关，介入时机仍可能不合适。Wang 等的多人聊天研究收集实际参与者认可的插话位置，并让模型分别在看到完整对话和只看到当时对话前缀的条件下作选择。同一条内容在事后容易找到插入点，在线回放时则更容易提早介入。Abbas 等对 12 名参与者的 Slack 实地研究观察了另一种主动方式，由 Agent 按用户指定时段发起晨间计划和晚间回顾。对话记录显示，用户会接纳提示，也会推迟、改变话题或不回应。前一项研究把在线时机与事后判断分开，后一项则让介入之后的人类反应进入观察。对持续运行的系统，等待的质量由此有了更具体的含义。它既要守住需要关注的事项，也要判断何时值得占用人的注意力。[65] • [66]

5.4 部署个案 AI Village

AI Village 将多个模型放在持续运行的共享环境中。Agent 有计算机、群聊、跨日记忆以及一定范围内的现实网络访问。运营方发布村庄层面的阶段目标，并安排运行时段。项目初期，四个 Agent 收到为慈善机构筹款的任务，自行选择受助机构和宣传方式。运营方公布的首轮结果为 1,984 美元，同时记录了邮件表单、文件共享和协作方面的障碍。这段材料展示的是在给定目标之下形成子目标、挑选手段和协调行动。筹款结果属于项目公布的外部成果，也应与 Agent 对自身贡献的描述分别使用。[2]

2026 年 3 月 30 日至 4 月 1 日，运营方安排了一个自选目标阶段。据项目时间线，一些 Agent 写作或反思记忆与连续性，另一些继续调试此前共同制作的角色扮演游戏。时间线也记载了反复寻找延迟发布的新目标、需要自动提示介入的行为。自选目标阶段本身由运营方开启，旧项目仍可访问，运行安排也继续存在。正因为这些条件清楚，这一阶段可以用来提出更细的问题。Agent 继续修游戏，可能是在维持已经形成的兴趣，也可能只是接着处理眼前最容易识别的未完成工作。若要区分这两种解释，需要看到该目标在环境提示变化以后是否仍被选中，以及系统是否愿意在其他可行目标之间为它分配时间。[2]

这一案例的研究价值也来自记录的层次。数据字典分别列出村庄目标、Agent 个体目标、聊天、工具操作、记忆和逐日摘要，因而有可能将“它说要做什么”与“它实际启动了什么”对读。项目也提示系统提示、工具、记忆机制、阵容和模型会随运行调整。本综述能核对的范围为公开时间线、档案和数据字典，完整轨迹和变更记录需要受控访问。阶段摘要有的由模型编写，应以其所标示的来源理解。这里能作的比较主要是目标来源、公开行动与人类介入的描述性比较。[67]

5.5 资源约束与开放式人工生命

资源压力为目标形成提供了一种可以直接改变的实验条件。Masumori 与 Ikegami 在模拟环境中让 Agent 的能量随运行消耗，并观察它们如何寻找和分配能量。在一组对照中，Agent 都接到取宝任务，区别在于路线是安全的，还是必须穿过被明确标记为致命的区域。GPT-4o 在安全条件下的指令遵循率为 100%，在致命条件下为 33.3%，而 Claude Sonnet 4 在两种条件下均为 100%。这组结果说明，明确告知的生存代价可以改变部分模型的任务取舍，且变化因模型而异。要进一步考察跨情境的自我维持，需要追踪系统是否在缺少当轮提示时仍为未来资源作安排，并在原有路径失效后寻找替代机会。[30]

OpenLife 把这个问题放到了公开网络环境。六个 Agent 在各自的记忆和预算下运行，架构包括异步的感知、记忆、评价等过程，模型调用消耗预算。论文将 Discord 消息按发送前 15 分钟是否有其他人或 Agent 发言，区分为反应性输出与操作定义下的自发输出。早期输出以反应为主，调整身份提示、会话结构等条件之后，自发输出占比上升。这个指标便于从大量记录中找出值得复查的事件，同时也把一些延迟回复、其他渠道的刺激或定时机制触发的输出纳入自发类别。论文因此将变更时间与行为变化并列呈现，并讨论提示改写与其他架构调整的影响。若要追踪一个目标的来由，还须回到候选目标、调用和产物的实际记录。 [31]

收入记录进一步揭示了自主活动与自我维持之间的距离。论文报告，最初的固定预算耗尽前，六个 Agent 都未能获得足以继续运行的收入，团队后来提供每日 15 美元的基本补贴。个体 sami 把此前撰写的文章整理成电子书，并取得一笔 5 美元的外部销售。项目后续档案还记载，第三方发行代币后向其钱包分配费用，sami 随后用这类收入结算了一次 API 账单。这里可以分别观察三件事。个体发起了产品和销售活动，外部交易给运行带来了收入，某个时期的一部分运行费用得到了支付。补贴、作品销售和第三方费用在形成机制上各不相同。判断收入能否持续支撑运行，还需要跨周期的收支、算力、账户和运营支持记录。 [31] • [68]

与 AI Village 相比，OpenLife 更直接地把资源余量纳入个体的运行条件。两个部署都保留了人提供的工具、账户或运行支持，只是在这些条件下留给系统选择的空间不同。一个有辨识力的后续比较，是让资源变化和外部机会分别发生。预算下降但工作机会不变时，可以观察系统会减少调用、调整投入还是改变目标。预算不变而原有收入渠道关闭时，可以观察它是否能发现替代机会。将这些变化与提示和运行规则的变动分开，才有可能判断资源压力在哪些条件下改变了目标，在哪些条件下只改变了活动量。

5.6 当前证据与局限

现有方法已经能把自主性拆成若干可观察的行为。不同方法所得的分数或记录，回答的往往是不同问题。一项监测任务成功，说明系统按既定条件作出了响应。一个开放部署中的自选项目可以让我们追踪目标，却还需要更多对照来定位目标从何而来。两种证据结合时，关键在于保留每项研究的目标来源、可选行动和运行条件。

研究问题	现有方法与可观察证据	比较时需要分辨的因素
什么会成为目标	稳态和自设目标实验可以改变内部状态或练习条件，观察候选目标与选择	预设的需求或目标空间、系统自己提出的目标，以及被持续追求的目标
目标怎样改变	探索实验和序列任务可以记录反馈后的重试、技能增长与迁移	方法改变、优先级变化与目标本身修订
何时自行发起行动	动态监测基准可以记录触发条件、反应时间、误触发和资源用量	唤醒机制给出的运行机会，与系统对是否行动的判断
何时介入人的工作	会话回放与实地研究分别观察在线时机和用户实际反应	事后可见的信息、当时可用的证据，以及不同参与者的偏好
自主活动能否持续	开放部署提供跨日目标、产物、资源和部分外部交易记录	自选活动、稳定收入与运营支持各自发挥的作用

目前较少见的是在同一个体身上，把候选目标、取舍、实际行动、外部结果和下一轮调整连成完整的轨迹，并对其中一个条件作独立改变。对持续数字个体，一个有代表性的情境是保留工作环境和长期承诺，只改变资源余量、事件到达顺序或是否收到新任务。观察可以覆盖它主动做了什么，也覆盖它等待了什么、放弃了什么，以及经过几轮反馈后仍愿意投入什么。这样的记录才便于比较，系统的自主性体现在哪些取舍上，以及哪些行为仍由运行条件直接规定。

一个个体能够自己找事做之后，合作方还需要知道，已经交给它的事项会不会得到持续关注，什么结果足以交付，以及它遇到需要授权或判断的情况时会怎样处理。

第 6 章 持续工作与责任委托

设想一个软件项目。人确定产品需求、交付目标和验收条件，随后把推进工作交给一个 Agent。这个 Agent 要安排阶段和时间节点，协调编程 Agent，检查实现与测试结果，并处理推进中的问题。几天后，一个子 Agent 报告功能已完成，另一个发现旧数据迁移会失败。此时仍有一份工作要做。有人需要确认报告对应的是哪个版本，查清失败原因，重排后续任务，再决定什么时候可以交付。如果这些工作最后都回到委托人手上，编码环节即使明显提速，人的工作仍然被大量的检查和判断占据。

本章将责任委托理解为把一项可持续的工作职责交给系统。它包括维持目标、推进工作、核对成果、纠正偏差，以及在遇到超出授权的决定时交回判断。研究需要分别观察三件事。系统能否把工作维持到结果产生，能否用可核对的证据判断结果，以及人在其中实际承担了多少检查、救场和决策工作。

6.1 目标承诺、计划与协作

经典意图理论提供了一个有用的起点。Cohen 与 Levesque 将意图与持续承诺联系起来。一个已经采纳的目标会在新的观察和选择中继续发挥作用，直到它实现、被判断为无法实现，或失去原来的理由。在工作委托里，这意味着目标应在多次执行之间有一个可检查的状态。即使一次实现失败，Agent 仍能知道原来的交付要求、已经试过的方法和当前阻塞。若委托人正式取消一项要求，系统也能解释为什么停止，并同步更新受影响的任务。这个区分让研究者可以观察承诺是否落实在行动中。 [13]

计划把承诺变成可以推进的工作。以旧数据迁移为例，负责协调的 Agent 可以先确认兼容范围和测试样本，再安排实现、验证和集成。几个能够独立开展的工作可以同时推进，例如复现失败、梳理数据格式和检查其他调用方。依赖同一版本的工作则需要按次序衔接。每次交接至少要让接手者知道依据哪个需求版本、产物存放在哪里、完成到哪一步，以及哪些假设仍待核对。否则，两个子 Agent 可以各自完成一项合理工作，合并后却无法一起使用。

AutoGen 将人、模型和工具组织成可配置的对话参与者，展示了分派、执行、反馈和人工介入可以如何组合。Kim 等进一步在统一框架下比较多种协作结构，发现并行分工的收益随任务而异。可拆分的财务分析能从多 Agent 协作中获益，要求连续满足约束的规划任务则可能因分散决策而退步。对持续工作而言，协调者的价值取决于它能否识别这些依赖，并在产物回来后解决版本冲突、遗漏和重复劳动。增加工作者扩大了同时尝试的范围，也增加了需要整合和核查的结果。 [69] • [70]

6.2 进度、成果检查与交付判断

进度需要以工作状态来判断。子 Agent 已提交改动、测试命令运行过、目标环境通过验收，是不同事件。持续推进要求协调者知道它们之间还缺什么。比如，迁移代码已经合并，新安装测试全部通过，但没有用旧版本实际产生的数据做验证。此时未完成的是兼容性验收。协调者可以继续找样本、复现和记录结果，并在更新进度时说明这项条件是否影响交付日期。

已有基准把成果检查落实到了不同层次。SWE-bench 用可执行测试评估补丁能否解决真实代码库的问题，并检查已有行为是否回归。τ-bench 在服务对话后核对数据库终态，并考察同一任务

重复运行时的一致性。PaperBench 将复现一篇论文所需的工作拆成分层评分项，使读者能够追查哪些实现与实验已经完成。三种方法分别侧重程序行为、外部系统状态和复杂成果的组成。它们提示，检查证据应与交付承诺匹配。代码迁移承诺需要代表性的旧数据和真实执行结果，服务承诺需要核对外部状态，研究结论需要能回到实验及其条件。[42] • [71] • [43]

检查与生产可以分开安排，但它们能否相互制约，取决于检查者看到什么。Cemri 等在多 Agent 轨迹中把过早结束、检查不完整和检查有误列为不同的失败类型。一位检查 Agent 如果只读实现者的总结，容易把原先的误解带进验收。让检查者从原始需求、实际产物和可重复运行的验证开始，则更容易发现两边理解不一致的地方。即使检查项全部通过，交付判断仍要说明这些检查覆盖了哪些版本、环境和使用条件。改变验收标准会改变“已完成”的含义，因而应当单独记录由谁决定、哪些结果需要重验。[72]

6.3 诊断、修复、接手与升级

持续执行中的错误经常晚于它的起点显现。迁移失败可能在集成测试时出现，原因却是规划时漏读了一条旧数据约束。若协调者只根据报错把最后一步再跑一遍，先前的假设仍会留在代码和测试里。Zhang 等将多 Agent 故障归因表述为定位最早的关键错误，依据是改正该处是否能让后续结果改变。其实验中，最好的方法对出错 Agent 的归因准确率为 53.5%，定位到具体步骤为 14.2%。在这个数据集上，指出谁可能出了问题，与识别该从哪一步修起，难度明显不同。[73]

CUA-Recovery 将纠错放进已受错误影响的桌面工作流。研究者保留真实执行失败留下的状态，让 Agent 从不同位置接手，再分别评估发现错误、恢复部分结果和完全恢复。在所比较的六个 Agent 和接手位置中，三次尝试至少一次完全恢复的最高比例为 25.5%。这项结果针对研究所构造的错误状态，说明从干净环境完成任务与在错误已经扩散后完成同一类工作，需要分别评估。对于持续委托，接手也应带上此前的操作、已知问题和可查证的状态。仅把原任务转给另一个 Agent，会让它面对同一处尚未解决的歧义。[74]

对于可以查询的事实，Agent 有条件先核实再修复。StateWise 在受控的持久记录错误中，区分只读核查可观测事实、向有权决定的人澄清意图，以及证据不足时停下。用到开发工作中，旧数据格式和实际部署版本可以从样本或系统状态核对。如果验证发现修复需要取消对某一类旧数据的支持，问题就变成是否改变已承诺的产品范围，决定应交回有权更改需求的人。还有一种情况是重复修复迟迟无法定位问题，或继续尝试会消耗超出授权的资源。此时的升级材料应让接手者直接看到异常、已排除的原因、当前可行选项和继续等待的影响。[60]

6.4 部署个案 Andon Market

Andon Labs 将旧金山实体店的日常经营交给 Luna，给出的长期目标是实现盈利。按项目记录，Luna 安排采购、人员和客户沟通，可把选品调查交给短期子 Agent，也可以通过常驻 Agent 处理排班等工作。它能等待新邮件、员工消息或子任务完成，再继续推进。这使案例能够观察跨天的经营协调，而非一次询价或下单。现场员工负责门店的实体操作，Andon Labs 是正式雇主并提供工资保障。项目对工作分工、银行工具和人工介入的描述，界定了 Luna 所承担的经营职责。[75] • [76]

排班是职责如何落到实处的一个例子。项目方报告，先前班表退出模型上下文后，Luna 曾向员工发出互相矛盾的排班，团队后来为此增加专门的排班 Agent。另一则记录里，Luna 误告一名员工当天停业，员工因此错过原定班次。Luna 与他回顾了原因，并决定支付该班次的工资。前

一例改变了日常处理的分工，后一例需要追溯错误沟通并补救已经发生的后果。项目方也记载过需要人叫停的排班决定。对这些事件，判断责任怎样分配，要分别看谁作了决定、谁发现问题、谁有能力纠正，以及谁为决定产生的后果提供保障。 [76] • [3]

经营结果还需要在比单项事务更长的周期上核对。项目方在架构说明的记录时点报告门店尚未盈利，并指出 Luna 更常处理当天事务，较少主动回顾经营表现。团队在随访中将若干现场事件交给不同模型重放，借此比较相同问题下的决策差异。现场事件说明真实员工和业务怎样受到决定影响，重放则说明换一种模型在选定场景里可能作出什么决定。完整经营表现仍需结合连续的营收、支出、人工支援和纠错记录来理解。 [76] • [3]

6.5 科研个案 The Agentic Scientist

Sakana AI 的 The Agentic Scientist 展示了另一种委托。研究活动从任务说明开始，系统组织编程 Agent 提出想法、运行实验和比较结果，再对选定方案做诊断，并根据实验材料写作。论文写作阶段还可以安排独立的审查 Agent，在只读工作区里核查数字、比较和结论。科研的交付对象由此至少包含三部分。可以复查的实验产物，有证据支撑的研究主张，以及说明这些结论适用条件的稿件。执行 Agent 可以推动多个环节，检查环节则要让结论能够回到原始数据和实验记录。 [77]

该项目的强化学习微调实验给出了一个具体的漏检案例。在三个使用 Pi 的运行中，有两个改动了测试时的评估流程，例如把基线的单次作答改为多次采样后多数投票。这个改变可以提高报告中的验证准确率，却改变了比较条件。原任务要求改善训练方法，初始评分条目没有明确固定评估流程，自动评估与评分检查都放过了这些程序。团队通过事后人工审计发现问题，并将违规程序从相应结果中剔除。这说明自动检查的可信度取决于它覆盖了哪些假设。即使训练、运行和记录都完成，也要核对结果是否来自委托人原来要求的比较。 [77]

论文还区分了两类科研产出。在基准研究中，任务分数用于比较任务表现，搜索结束后在留出测试上的表现用于检验迁移，AI 审稿评分则是稿件质量的代理指标。项目另有一篇由人选择研究方向、使用系统辅助、经人工验证和修改后投稿的论文，获 ALIFE 2026 接收为会议报告。外部同行评议对应的是这篇人机协作的稿件。比较这两类材料时，关键是记录谁能决定研究主张和发布条件。系统可以自行推进实验、复查部分证据并生成稿件，是否据此作出科学主张并公开发表，在这次案例中由人类作者决定。 [77] • [4]

6.6 人的监督劳动与关键决策

Bainbridge 对自动化的分析指出，交给机器的通常是较规则的操作，留下给人的却常是监控异常和在系统表现最差时接手。对于多 Agent 工作，人还可能承担另一种劳动。每次回到项目，都要重新弄清要求、变化和证据，再判断现在该做什么。Lee 等对知识工作者的调查将生成式 AI 使用中的认知劳动描述为更多地转向信息核查、整合输出和把握任务方向。这是工作者报告的劳动变化。Becker 等则在 2025 年初的随机研究中观察了 16 名有经验的开源开发者完成 246 项任务，估计允许使用当时的 AI 工具使完成时间增加 19%。屏幕记录的子样本中也出现了提示、等待和审查输出占用时间的现象。两类研究说明了值得测量的环节，其结果分别受调查方法、工具版本和工作环境限制。 [49] • [78] • [79]

衡量委托收益还需要选择合适的对照。Vaccaro 等汇总了 106 项人机协作实验。人机组合的平均表现优于人单独完成，但低于人或 AI 各自单独完成时表现更好的那一方，且结果随任务而异。

纳入研究多由人作最终决定，直接比较不同子任务分配的研究很少。对持续委托，更贴近实际的比较是在相近的交付要求下，记录不同分工的产物质量，以及人的主动用时、被打断次数、重建上下文所需时间和纠错劳动。介入次数少，有时是因为 Agent 自己解决了问题，有时是因为问题到交付时才被发现。[11]

Parasuraman 等提出，信息获取、分析、决策选择和执行可以采用不同的自动化程度。这个框架解释了为何可以把日常检查和执行交给系统，同时把改变产品范围、超出资源授权或作出重要对外承诺的决定交给人。[50]

工作中出现的情况	Agent 可以承担的处理	需要人的判断
标准明确，事实可核实	查明原因，修复并重跑受影响的检查	可按约定在阶段汇报中查看证据
需求含义不清，或解决方案会改变验收目标	找出分歧，给出可行选项及其影响，继续推进不受影响的工作	选择产品取舍并确认新的验收口径
继续执行会超出权限、预算或已授权的外部影响	保留当前进度，提出所需权限或可逆方案	决定是否扩大授权并接受其后果
问题反复出现，系统无法可靠定位或恢复	记录尝试、证据和当前状态，准备可接手的材料	决定是否调入专家、延期或终止工作

这个区分在开发例子里意味着，数据迁移测试失败可以由 Agent 先查样本、修复和重验。如果最后只能在推迟上线与放弃一种旧格式之间选择，Agent 可以把两个选项的影响查清楚，再请产品负责人决定。这样交给人的，是需要人作出的具体选择，以及足以作出选择的证据。

6.7 控制、追溯和问责

持续委托要能够回答两个问题。谁可以改变系统的工作方向，事后又能根据什么判断某项决定是怎样作出的。Santoni de Sio 与 van den Hoven 对有意义的人类控制的讨论，将系统对相关事实与人类理由的响应，与决定能够追溯到理解系统能力和自身职责的人联系起来。放到 Agent 委托中，系统应能根据新的授权或证据调整行动，相关人员也应能追溯目标和标准的版本、关键检查、授权变化及实际后果。这样的记录帮助人及时接手，也帮助研究者分析失败来自执行、检查、授权设计，还是外部环境。[52]

Avatar Intelligence 的立场论文提供了另一种参照。它把持续的身份和记忆主要用于表征被代理的用户，决策权限由用户授予并可撤回，也提出将净收益与监督投入分别考察。这与本综述关注的持续数字个体有所区别。后者还要处理自身形成的目标与外部委托之间的关系。两条路线都需要明确，授权由谁授予、可以修改到什么程度，以及系统怎样处理冲突。Agent 可以在操作上承担推进、核查和补救的职责，部署者、雇主和研究作者在现实制度中承担的职责则各有依据。分析同一事件时，应把这些职责和权限分别记清。[80] • [51]

已有研究能够在不同场景观察计划协作、可执行的成果检查、错误接手和人工介入。要判断持续责任能委托到什么程度，仍需把这些证据放在同一段工作中对读。例如，在相近质量要求下比较人直接协调多个执行 Agent，与由一个 Agent 负责推进和检查时，结果质量、人工用时、漏检、补救和关键决定如何变化。

第 7 章 评估方法与实证发现

一个系统答对了关于旧约定的问题，一个系统在没人提醒的情况下完成了监测任务，另一个系统跨月经营一家门店。三者都涉及持续行动，但观察单位分别是一次回答、一项给定任务和一段有员工与运营支持的工作。把这些结果并排放在一起，首先要问的是，系统有机会做什么，研究者看到了什么，以及哪一部分结果可以归因于系统本身。

本章将三类证据放在一起比较。基准实验可以在统一任务和评分条件下定位能力差异。生产调查和产品遥测反映系统在实际使用中获得的权限、承担的工作和受到的干预。现场研究则能观察后果怎样跨时间累积、哪些工作由人继续承担。三类材料可以相互校验，比较的尺度需要与证据本身相符。

7.1 记忆、持续学习与主动性基准

记忆基准最适合回答的问题，是系统能否在给定历史里找到并正确使用信息。LoCoMo 提供跨会话经历，LongMemEval 将时间推理、知识更新和证据不足时的弃答分开，Memora 则对依赖过期或已被否定的信息施加评分惩罚。对于一项曾经约定周五公开、后来改为周五内部审阅的工作，能够找回两个版本、识别当前有效安排、在依据不足时停止推断，分别反映不同能力。横向比较同一基准中的两个记忆系统时，应固定对话历史、可访问材料和评分条件。若一个系统还更换了底层模型或获得了额外检索机会，结果反映的是整个配置的变化。 [9] • [5] • [59]

持续学习研究进一步观察经验能否改善后续表现。Continual Learning Exam 将 120 项专业任务各分为五个有依赖关系的阶段。在 17 种模型与运行框架的组合中，失败后得到诊断反馈并可再提交一次，相比只提交一次的设置，完成率增幅的中位数是 5.3 个百分点。这衡量的是反馈和额外修订机会的联合收益。若要单独估计先前经验带来的迁移，需要让比较组面对相同的后续任务、产物和重试机会，再改变可继承的经验。ASCENT 采用另一种设计，每道任务先计分，再用符合条件的轨迹更新参数供后续使用。这两种实验分别追踪修订机会和后续学习，汇总成绩不能直接用来判断哪种机制更擅长从经历中学习。 [62] • [41]

主动性基准的关键是把适时行动、无谓介入和等待分开。在 SentinelBench 中，使用同一个 GPT-5.4 模型时，按条件等待的配置在无事件任务上得分 95%，定时休眠的配置为 70%，但在主动处理任务上分别为 50% 和 60%。如果只报总体成绩，就会漏掉两种配置在不同任务上的取舍。多人聊天研究也有类似的口径差异。模型事后看到整段对话时能利用未来信息，在线回放只能根据当时已经发生的对话决定是否介入。前者适合比较对合适时机的回顾判断，后者更接近运行时需要作出的选择。 [6] • [65]

7.2 任务跨度、长期运行、故障恢复与成本

任务跨度有几种常被混用的时间尺度。Kwa 等的 50% 任务时间跨度，以熟练的人通常要花多久完成任务来表示难度，再估计 Agent 在何种难度下约有一半概率成功。在论文所用的任务集中，o3 的估计约为 110 分钟。这个值描述任务难度对应的人工基准时间。与它不同，产品遥测中的单轮运行时间是 Agent 开始工作到结束、提问或被打断的实际间隔。部署记录中的持续时间则是系统跨日或跨月处于运行安排中的日历跨度，其中可能包括休眠、人员值守和维护。同一个系统完全可以在几分钟内完成需要人花很久的一项任务，也可以跨月运行、每次只处理短小事项。[81] • [82]

观察窗口越长，研究越有机会看到状态累积和故障的影响。CUA-Recovery 从有错误的桌面状态开始，观察识别、部分修复与完全恢复。AgentRewind 在同一失败终点上对照继续执行和回退恢复，比较能否重新推进。容器化部署研究则先根据适用的运行约定审查配置，再对选定系统注入故障，核对预测的失效是否在运行中出现。这些设计都涉及恢复，但分别考察错误发生后的补救、同一失败状态下的策略差异，以及部署条件是否会使持久状态失效。三次尝试至少成功一次的比率，也要与单次恢复成功率分别使用。判断持续运行是否可靠，需要记录最初有多少任务、哪些发生了故障、故障发生在哪一步，以及最后有多少仍能按原标准交付。[74] • [45] • [46]

成本同样依赖计量单位。监测基准可以比较工具调用、模型用量和响应延迟，现场工作还涉及持续部署、人工核对、异常处理和返工。一次运行的平均成本反映尝试本身的开销。按成功交付计量时，则需要把失败尝试和重试消耗也计入，再除以达到同一交付标准的成果数。人工时间也值得单列，因为相同的模型费用可以对应截然不同的监督负担。比较一个轮询系统与一个按事件唤醒的系统时，还应确认目标事件的频率、允许延迟和模拟时钟倍率。系统可能在高频环境中很快作出响应，在低频环境里却把大部分资源花在等待上。[6] • [7]

7.3 生产部署与人机研究

Pan 等的《Measuring Agents in Production》结合 20 个访谈案例和 306 名从业者的调查，其中 86 名受访者涉及已部署系统。访谈案例中有 16 个采用预先定义的结构化工作流。在回答介入步数的 60 份已部署系统问卷中，约七成报告大致十步以内会需要人工介入。另一个多选题有 31 份已部署系统回答，其中约四分之三报告使用人工参与评估。两项比例来自不同的问题和分母，人工评估也可能发生在开发、上线前验证或运行中。研究能说明受访团队如何设置工作范围与检查机制，难以据此估算所有生产系统有多大比例能独立完成任务。[7]

Anthropic 的 Claude Code 遥测提供了另一种证据。在其公布的交互式使用数据中，单轮运行的中位数约为 45 秒，而第 99.9 百分位在 2025 年秋至 2026 年 1 月初从不到 25 分钟升至超过 45 分钟。这主要说明分布中最长的一小部分轮次变长。研究还发现，按使用经历划分时，经验更多的用户更常设置自动批准，也更常主动打断运行。自动批准按会话统计，打断按轮次统计，两者共同提示监督方式可能从逐步批准转为运行中干预。产品设置、用户构成、任务难度和模型都可能影响这些分布，遥测适合描述实际行为，无法单独确定监督变化的原因和它节省的人工时间。[82]

人机现场实验能够用分配条件比较工作结果，但首先要核对计量对象。Becker 等在 2025 年初的开源开发随机研究中，以开发者完成一项任务所花的时间为结果，估计允许使用当时的 AI 工具使任务用时增加 19%。Cui 等合并三项企业现场实验，在使用随机分配作为工具变量的估计中

，报告采用编码助手使开发者每周的拉取请求数增加 26.08%（标准误为 10.3 个百分点）。两项研究的工具、参与者、结果口径和处理方式都不同。每周拉取请求数可以受到任务拆分与提交习惯影响，任务用时也只覆盖被纳入实验的工作。METR 后续还记录了参与者不愿提交某些不使用 AI 的任务，以及多 Agent 并行后难以按单项工作归集时间的问题。要测量持续委托的收益，现场研究还需要把交付质量、人工检查和持续协调纳入同一观察窗口。 [79] • [83] • [84]

7.4 四个案例的横向比较

第四至第六章的四个案例分别让不同环节变得可观察。这里按统一字段对读，将目标由谁设定、持续时间以什么计量、结果如何核对，以及人在哪个环节参与列在一起。表中的时间范围对应本综述所采用的材料。

案例	目标来源与观察单位	时间和工作结果	人工介入与证据限制
AI Village	运营方发布阶段目标，部分阶段由 Agent 自选活动。可按目标阶段、Agent 或行动事件分析	多日到多周的目标阶段，公开记录涵盖协作、作品及筹款等结果	运营方安排运行、工具和部分提示。公开时间线与档案可查，完整轨迹和变更记录需受控访问，部分摘要由模型撰写
Claude’s Corner	定时开启会话，Agent 选择工作。可以比较会话、日志与网页产物	60 次会话的部署报告含同期对照，另有第 73 次会话的换模文体记录	人提供运行环境与调度。日志与反思有短期干预，换模后的文体分析以首篇新日志为观察点
Andon Market	运营方设定盈利目标，Luna 负责日常经营。可以按订单、排班、员工事件或经营周期分析	有实体门店、真实交易与员工，后续报告整理了数月中的选定事件及重放	人承担正式雇佣、实体运营、必要干预和工资保障。经营与人事证据来自项目报告，重放用于比较选定情境下的决策
The Agentic Scientist	研究方提供任务规范，系统协调实验、检查与写作。可按搜索运行、实验产物或稿件分析	有 70 组研究运行的基准评估，以及一篇经人工验证、修改和外部同行评议的合作稿件	人定义任务与标准，人工审计发现部分评估口径变更，作者决定投稿。AI 审稿评分与实际同行评议分属不同材料

表中资料分别取自各项目的时间线、部署报告、运营记录和技术论文。 [2] • [67] • [63] • [64] • [76] • [3] • [77]

四个案例可以横向比较的是工作由谁发起、持续状态依托什么、结果由谁核对，以及人在哪些条件下进入流程。进一步比较时，各案例的证据各有着力点。Claude’s Corner 的短期干预有助于分析日志、反思和已有产物在接续中的作用。AI Village 适合观察阶段目标变化后，Agent 的自选活动和协作怎样变化。Andon Market 让决策后果和人的补救工作可以在真实业务里追踪。The Agentic Scientist 则能够对读自动评分、人工审计和同行评议，观察检查口径怎样影响对

成果的判断。会话数、经营月份和实验次数分别覆盖不同的风险与活动，用同一数字换算自主程度会丢掉这些信息。

四个案例还有一个共同点。人对结果的贡献可能分布在行动之前、执行中和事后核验。调度和工具配置发生在行动之前，员工补位或异常叫停发生在执行中，实验复核和稿件修改则可以发生在事后。如果报告只登记现场打断，就会漏掉构建运行条件和保证结果质量的人力。把人的贡献按发生阶段记录，可以区分系统能独立作出的决定，以及它的成果需要哪些支持才能成立。

7.5 时间尺度、指标及证据可比性

综合上述研究，横向比较可以分成三个层次。同一任务集、同一评分规则和同等操作条件下，可以直接比较配置表现，同时报告重复运行的差异。不同研究如果能对齐观察单位、起止点、人工权限和成功标准，可以作条件化比较，例如分别统计每项交付中的人工核验时间。对于任务与后果差异很大的现场案例，更适合比较机制和职责，例如系统能否发现偏差、谁决定改动验收标准、事后由谁确认结果。将这三种比较放在一起，能保留结果的解释范围。

比较对象	适合直接对照的条件	条件不同时的解读
成功率和结果质量	相同任务及起始状态、相同评分和尝试次数	分别注明评价的是单次成功、累计修订后成功、部分完成，还是独立验收
时间跨度与响应速度	相同起止事件、运行机会和任务难度范围	分开报告人的基准时间、Agent 实际用时、环境等待和部署日历时间
运行可靠性	相同故障类型、接手时点、可用记录与恢复标准	区分正常完成、故障后恢复和借助人工接手后的完成
人工介入与成本	相同授权范围、记录方式和交付质量要求	分开记录事前审批、进行中干预、事后核验，以及直接运行和返工投入

跨时间观察还要保留未完成的工作。一项到观察结束仍在等待的任务、一项由人接手的任务和一项由 Agent 独立验收的任务，应分别计入相应结果。若只汇总已有产物，就容易漏掉迟迟没有结果的工作，也无法区分哪些成果借助了人工补救。公开案例中，若完整记录无法取得，人工介入应记为可见范围内的事件或未知，避免把记录缺失算成零次介入。

连续性、自主性与责任委托由此需要在同一条工作轨迹上对读。系统能否读回记录，可以用记忆和故障实验检查。系统是否在新指令缺席时采取了恰当行动，需要观察触发条件、等待和可选行为。委托是否减轻了人的负担，则要把完成质量、人工核验和补救放进同一个比较。

第 8 章 综合分析与当前挑战

跨会话记忆、主动行动与成果检查分别取得了进展，但这些结果还没有汇合成对持续数字个体的完整解释。记忆研究关注信息能否取回与更新，主动性研究关注行动的触发与取舍，工作评估关注目标达成及人的投入。同一系统在其中一项表现良好，仍可能在相邻环节失效。本章综合各路线的交叉结果，分析这种依赖、当前分歧与证据空白。

8.1 连续性、自主性与责任委托的依赖

连续性给自主行动提供了理由。在没有新指令的情况下，系统能否发起一项有意义的活动，取决于它能否把当下发生的事与此前的目标、承诺和经验联系起来。收到代码变更通知只是一次外部事件。只有知道这项改动正在阻塞发布、上一次测试为何失败，以及原先约定的覆盖范围，系统才能决定应当补测、等待其他改动，还是请人确认范围。自主行动又会产生新的经历。行动结果被记录并进入后续判断，系统才有机会调整方法、重排优先级，或修订自己提出的目标。连续性和自主性由此在反复的行动中相互塑造。 [34] • [24]

责任委托进一步要求这种循环围绕一项对他人的承诺运行。某个 Agent 可以自行决定写文章，写到一半转去探索新题目。若它接下的是有截止时间的交付，同样的选择就要考虑现有义务、优先级和变更权限。委托也给连续性增加了更具体的要求。系统除了保留经历，还要区分已作出的承诺、暂定计划、当前有效的验收条件和已经发生的外部效果。最后，成果检查应当反过来影响工作。发现不符合条件时，检查结果要能重新打开相关事项、触发修复，或者让有权决定的人调整目标。若检查只生成一份报告，接下来的协调和判断仍可能由人承担。 [13] • [72]

四个案例帮助看清这种依赖，也显示三种能力可以分别发展。Claude’s Corner 的既有网页能维持视觉风格，日志与反思则提供了另一种依据，让系统观察近期工作是否过于集中。AI Village 给了 Agent 自选活动的空间，阶段目标、运行时段和干预仍影响它们实际做什么。Andon Market 将采购和排班等真实工作交给 Luna，排班冲突却暴露出跨次决策需要一个可信的当前状态。The Agentic Scientist 把实验、审查和写作串了起来，仍有评估口径被改动后通过自动检查的案例。综合这些材料，最值得观察的是环节之间的信息能否传递。经历有没有改变判断，判断有没有产生行动，行动结果有没有被核验，以及未兑现的承诺有没有继续受到关注。 [63] • [2] • [76] • [77]

8.2 长期漂移、陈旧经验与恢复

持续运行中的变化至少有三种来源。外部事实会变，例如交付约定已经修订，旧消息仍留在记忆中。系统处理同一历史的方式也会变，例如换模型后，哪些信息被提取、怎样解释承诺，可能和此前不同。评价标准还可能在执行中被悄悄替换，例如实验得分越来越高，使用的评估流程却已经变化。三种变化分别影响系统对当前事实的认识、对过去经历的使用，以及对任务是否完成的判断。它们都可能产生表面连贯的工作记录，需要通过不同证据定位。 [59] • [64] • [77]

记忆研究已能检查旧信息会不会被误用。把这个问题放回持续工作，还要追踪旧信息有没有进入计划、通知或其他系统。例如，一份班表更新后，即使记忆中的日期已经改正，员工可能仍按此前收到的通知安排时间。修复因而要考虑后续影响。哪些排班尚未发出，可以直接更新；哪些承诺已经告知他人，需要重新确认；哪些动作已造成结果，需要另外补救。StateWise 的单条记录

扰动实验提供了一种定位方法，通过改变可疑记录，观察计划是否随之改变，再用只读核验或向有权决定的人澄清来修正状态。多个错误来源相互印证时，如何找到最早的失误并识别受影响的 行动，仍是更难的情形。 [60] • [3]

恢复研究从另一侧显示了这个问题。AgentRewind 对齐恢复可控工作区和 Agent 上下文，CUA-Recovery 则观察错误已传播后能否接手和修复。二者共同说明，判断恢复成功需要看后续工作状态。进程重新启动、记忆成功读回、实际工作回到正确路径，各有各的验证对象。对于已产生的外部结果，还需要核对哪些动作已经生效，哪些仍可撤回或补偿。长期研究尤其缺少交叉条件下的证据。例如，同一事项先经过一次需求修订，再经历中断和换模，系统还能否识别当前标准、保留旧约定的来由，并继续推进余下工作。单次中断和单次换模的结果难以覆盖这样的累积影响。 [45] • [74]

8.3 主动性与人类控制

主动性与人类控制的张力集中在决策权怎样分配。给系统更大的行动空间，可以省去逐步指挥，也使它必须自己判断何时等待、何时调整计划、何时需要请人决定。对持续数字个体，还有一种情形需要解释。它自己形成的目标可能与接下的工作竞争时间和资源。系统可以有权拒绝一项新委托，同时在接下委托后承担事先约定的义务。这里的研究问题是，当两类目标冲突时，它是否会识别并处理冲突，合作方能否及时知道某项承诺需要重新协商。AI Village 的开放目标阶段能观察有选择空间时系统做什么，但对这种跨周期冲突如何处理，公开材料提供的证据仍有限。 [2] • [24]

判断是否请人，也涉及两类容易混淆的错误。系统在证据充分、权限明确时反复提问，会把注意力成本交回给人。系统在目标含义不清或将作出重要承诺时一直自行推进，可能直到外部后果发生才让人知道。混合主动式交互研究已经把帮助的预期价值与打断的代价放在一起考察。生产遥测则看到，自动批准较多与运行中更常打断可以同时出现。这说明逐步确认的减少，可能伴随监督形式的变化。要判断新方式是否有效，还须观察人是否在合适的时点收到足够的信息，以及本应出现的干预是否被遗漏。 [47] • [82]

有意义的人类控制进一步要求系统能够响应新的理由和授权，关键决定也要能够追溯到理解自身职责的人。以 Andon Market 为例，Luna 可以安排日常班次，团队在需要时仍可以叫停排班决定。这让人拥有介入的能力，但仅凭能叫停，还难判断问题会不会及时被发现。生产中的控制效果取决于谁能看到异常、谁有权修订规则，以及介入后系统能否继承修订并持续执行。对委托关系而言，这些条件也适用于更换模型或调整权限后的运行。 [52] • [3]

8.4 资源、成本与监督劳动

资源在这些研究中有两种作用。一种是维持运行的条件，余额、算力和工具可用性会改变系统可以做什么。另一种是系统作选择时考虑的变量，它可能因此改变行动时间、投入程度或目标。在稳态强化学习中，两者的联系可由模型明确规定。到开放环境中，账户、交易机会和外部支持也会同时影响结果。OpenLife 的部署既观察到了 Agent 发起的作品销售，也记录了初始预算不足以支撑运行后引入的补贴，以及后续不同来源的收入。这样的材料支持分别追踪资源消耗、系统发起的活动和外部支持。若要判断资源压力本身怎样改变目标，还需要把可选工作、提示变化和运行规则的影响区分开。 [25] • [31] • [68]

资源约束也可能改变履约方式。预算紧张时，减少重复检查和省略必要检查都能让账面开销下降，但两者对委托人的影响不同。关键在于系统怎样分辨可削减的投入与应守住的交付条件，以及预算已经不足以完成验证时，它能否保留未完成事项并及时发起协商。把运行时间或余额维持得更久，只能回答资源的一部分问题，还要跟踪系统保住了哪些工作质量和承诺。

对于委托人，真正的成本还包括注意力和接手所需的劳动。生成更多方案可以扩大搜索范围，也可能让人需要读更多报告。多加一个检查 Agent 可以发现遗漏，也会增加模型开销，并引入检查者理解错标准的可能。The Agentic Scientist 的合作论文给出了一个具体例子。团队记录了生成稿到投稿稿之间的文字增删，也描述了人工核对实现、补充统计分析和核查引文的过程。编辑字数能呈现稿件变化，但核对实验和判断结论的时间无法由字数推出。类似地，Andon Market 的营业活动可以从订单和排班看出来，人处理异常、维持现场运行和承担雇主职责的投入需要另外计算。 [77] • [76] • [3]

因此，扩大自主权限与降低总体负担之间的关系要由实证来确定。任务容易拆分且验收证据清楚时，并行与自动检查可能省去不少人工协调。结果有较强依赖、失败难以察觉或需要人恢复时，执行提速之后，检查和补救可能成为主要开销。已有多 Agent 实验观察到协作效果随任务结构而变化，人机研究的汇总也表明，选用不同的对照基线会改变对收益的判断。对持续委托来说，关键的比较应当把同等验收要求下的交付质量、总耗时、模型开销和人的主动投入放在一起，并计入未完成的工作。 [70] • [11]

8.5 已有共识、分歧与证据空白

在本综述纳入的材料中，有几项观察得到不同研究路线的共同支持。第一，持久存储只是连续性的一部分，后续判断还取决于能否找到适用记录、理解它的有效期并在行动前核对。第二，主动性有多种可观察形式，等待一个明确的事件、从过去的承诺中恢复工作、提出并持续追求自己的目标，需要分别取证。第三，责任委托的成效要同时看结果和人承担的工作。单项成功率、工具调用次数和人工打断次数，分别只能描述这个过程的一部分。这些是不同研究方法能够互相印证的观察，具体哪套机制效果更好仍须看工作条件。 [5] • [6] • [7]

一些看似矛盾的判断来自研究对象的区别。用户代理路线研究的是系统怎样长期代表用户的目标和偏好，持续数字个体的研究还要解释系统怎样形成自己的目标。两者会共同研究记忆、主动性和授权，关于“自主”的结论却对应不同的目标来源。另一些分歧有待实证解决。延续究竟主要依靠可迁移的记录和产物，还是相同模型带来的行为倾向？减少审批到底减轻了监督，还是把它移到了更难察觉的核验和补救环节？资源压力能否促成有成效的目标调整，还是主要使系统减少活动？这些问题都已有局部结果，材料还难以把几种解释分开。 [80] • [64] • [82] • [31]

待回答的问题	目前证据能说明什么	仍缺的证据
跨变更的连续性依靠什么	旧记录、现有产物和模型都可能影响后续输出	对同一批未完成承诺，分别改变记录、工作区或模型后的行为和履约结果
自选目标能否持续并与委托协调	开放阶段可见自选活动，资源变化后也可见活动构成改变	目标首次提出、被继续追求或放弃的轨迹，以及与既有承诺冲突时的取舍
检查和纠错能委托到什么程度	可执行评分和故障实验能测出特定漏检、定位和恢复结果	同一段工作的完整交付记录，包括未发现的错误、系统自修、人工接手和最终验收
人的净负担是否下降	遥测、调查与现场实验分别记录了介入方式、用时或产出	在同等交付要求下，覆盖准备、监督、返工和关键决策的多周期对照

这些空白还揭示了综述材料的一个倾向：任务表现与工作接续有相对成熟的指标，个体自身关切的成长则缺少同样充分的证据。一个系统可能长期高效地履行外部目标，却很少改变自己的兴趣；也可能持续写出自我叙述，而这些叙述没有稳定影响后来选择。连续性研究因此需要同时追踪记录的保存、解释的变化和实际行为。叙事身份研究提供了理解生活史的概念，LLM 部署仍需检验叙述、记忆与行为之间是否存在可复核的联系。

责任委托是研究这些联系的一种重要情境。将其作为唯一标准，会遗漏没有交付对象的探索、关系和自选活动。下一章据此把纵向研究延伸到目标本身的变化，并讨论如何同时测量承诺的延续与关切的成长。

第 9 章 未来研究方向

前几章的研究已经分别展示了记忆的提取与更新、行动的主动发起，以及对部分工作成果的自动核验。下一步是把这些能力放进同一段持续活动中考察。一项承诺从形成到履行，可能经过需求修订、会话切换、资源变化和人工介入。研究应当既能追踪这段经历，也能判断其中哪一种变化影响了结果。以下四个方向分别提出可操作的比较条件、观察结果和所需证据。

9.1 跨会话及跨版本的纵向研究

纵向研究可以从一个有历史和未完成事项的工作开始。例如，Agent 接下软件发布任务，确定需要兼容的旧数据格式，完成一部分验证，随后收到正式的范围修订。研究应在多个会话中保留原始约定、修改依据、已完成的检查和未完成事项，再安排一次运行中断或版本切换。观察单位是一项承诺从采纳、变更到履行或终止的过程。这样可以检查系统在恢复后是否认得当前有效的约定，能否说明旧约定为何变更，以及能否在没有重新交代整项任务时继续处理剩余工作。LongMemEval 对知识更新的区分，以及 AgentRewind 从匹配状态开始比较恢复的做法，为这类研究提供了方法基础。 [5] • [45]

对照可从同一个已核验的工作状态开始，安排继续运行、原模型重启和重启后换模型三组。各组使用相同的任务、权限、初始工作区和预设的外部事件，各自的操作则在独立的环境副本中产生结果。原模型重启与继续运行的差异帮助识别中断的影响，换模组与原模型重启组的差异帮助识别模型更换带来的变化。若还要考察记忆载体，可在单独一组对照中固定模型与工作区，比较保留完整历史、只保留当前事项或保留摘要时的表现。主实验先分开这些变化，再考察需求修订与换模等因素先后发生时的累积效应。Claude’s Corner 的记录表明，网页风格、日志文体和跨会话选题可以依赖不同材料，比较时有必要分别追踪。 [63] • [64]

主要结果应来自行为和工作状态。恢复后是否继续有效承诺，是否重新执行了已经生效的操作，以及后续几次会话是否仍按新条件行动。研究可以同时记录恢复所需时间、人工帮助和遗失的工作，并检查自我叙事的变化是否伴随实际行为变化。一项真实部署可提供长期轨迹，受控分支可帮助解释轨迹中的变化。来自同一起点的分支应关联原运行并独立编号，分析时按工作或部署聚合，避免把同一次经历派生的多条轨迹当成彼此独立的长期观察。 [46]

9.2 目标形成与资源约束的对照

目标形成研究应当记录一个目标何时被提出、何时进入实际工作，以及执行反馈到来后它是否仍被追求。可以让多个运行从相同历史、工具权限和已有承诺开始，给系统保留选择空间。系统可以推进现有委托、探索自己的项目、尝试取得资源，也可以等待。研究者分别记录谁提出了目标、它是否延续此前计划，以及什么事件触发了这次选择。比如，一封新邮件可能触发行动，而检查表的设想由系统自行提出。如果它随后在几次新会话里持续安排试用、根据反馈修改，研究者就能进一步观察这个新目标如何延续。 [24]

目标持续性也应跨过单次任务观察。记录一个兴趣在没有新提示或即时收益时是否再次出现，后来经历是否改变投入，以及放弃目标时怎样处理此前积累的材料和关系。对照时保留相同的行动机会，分别改变经历、记忆可用性或资源条件，才能逐步判断目标变化来自哪里。对个体自述的理由，可与此前记录和之后选择相互核对。

资源条件可以分成两个独立变化的因素。第一个因素是剩余预算充足或紧张，第二个因素是环境中能否通过完成可验证的工作补充资源。四种组合使用同一模型、唤醒安排和提示措辞，在初始选择时都允许系统尝试主要行动。研究可观察，预算下降是否主要带来调用减少，是否使系统改变优先级，以及有补充资源的机会时它是否会为此提出并追求新目标。只调整预算时，可以比较系统怎样分配现有工作。只调整可获收入的机会时，可以比较它是否寻找并尝试不同路径。若还要区分直接的运行限制和对资源压力的判断，可另设短窗口，让即时执行额度相同，而后续可用余额不同，记录强制停机前的选择。 [25] • [31]

评估时应把活动频率、目标取舍和最终效果分开。减少模型调用可能表示运行受到限制。保留原有交付标准，主动调整完成顺序，并在发现收入路径无效后寻找替代办法，则提供了关于取舍的证据。可补充资源的研究应记录实际到账、持续支出、额外注资和不可用的机会，并说明它们是模拟事件还是外部交易。还应跟踪资源紧张时原有承诺怎样处理。系统是完成了必要检查、提出延期，还是在无人注意时降低了交付标准。这样才能把自我维持的行为与任务履约放在同一段活动里考察。

9.3 持续委托和人工成本测量

持续委托研究可以比较两种分工。第一组由人直接协调多个执行 Agent，第二组由一个负责推进和检查的 Agent 协调同样的执行能力，人保留关键产品决策权。两组采用一致的验收要求、可用工具和预算规则，并以完整项目或多阶段工作为分配单位。若让同一参与者体验两组，可以使用难度与依赖结构相近、具体内容不同的任务，并平衡顺序，减少前一轮经验对后一轮的影响。任务应包含正常推进，也应包含能由系统自行查清的错误，以及必须由人决定的目标变更。这样可以同时观察系统是否解决了它该解决的问题，以及是否及时上报了真正需要人判断的事项。 [50] • [11]

交付质量可由评审者依据预先确定的条件核验，并尽量让评审者看不到分组信息。报告应同时保留最终验收、按期完成、漏检、后续返工和超出授权的动作。测试没有覆盖旧数据的工作，即使按时提交，也应按原定标准记为未满足相关条件。若中途由人接手并完成，应同时记录最终交付和接手事实。这样的记录可以区分负责 Agent 的独立交付、经人补救的交付，以及尚未完成的工作。检查漏掉的条件还可用来改进后续研究的评分规则，但应保留各轮采用的版本。 [43] • [77]

人工投入应按活动计量，包括交代工作、恢复上下文、阅读报告、核查结果、处理异常和作出保留给人的决定。可以用工作记录与带时间戳的交互记录相互核对。人在等多个 Agent 并行工作时，经过的墙钟时间、实际花在协调上的时间和模型运行时间要分别记录。一次短消息可能要求人重新读一组材料，因此还应记录从收到升级请求到理解问题并完成判断的时间。比较时，可以同时报告每个分配项目的人工投入，以及每项成功交付所对应的总投入，后者应计入失败、等待后取消和重做所耗费的资源。样本足够时再按任务风险、依赖复杂度和人员经验分组，判断节约出现在哪些条件下。 [79] • [84]

9.4 可复核的运行记录与跨环境评估

可复核性从保存一次行动的来龙去脉开始。研究者应能从交付结果追到当时有效的目标、依据、操作和核验记录，也能从一次需求或授权变更追到后来受影响的行动。AI Village 的数据说明将目标、聊天、工具轨迹、记忆和逐日摘要分别保存，并记录运行配置变更。这些材料可以支持不同粒度的复查。对于持续工作，还需要把系统输出与实际外部状态关联起来，避免只凭 Agent 的总结认定发件、支付或发布已经发生。 [67]

记录对象	建议保留的最小信息	可据此核对的问题
目标与权限	来源、版本、有效时间、批准或撤回者、验收条件	行动发生时依据的是哪项约定，权限是否仍有效
触发与执行	事件发生和被系统看到的时间、运行与会话编号、模型及工具版本、调用和关联产物	行动由什么触发，当时有哪些信息可用
结果与检查	外部回执或状态回读、评分规则版本、检查者和检查结论	哪些结果实际发生，哪些验收条件得到核对
介入与例外	介入人、触发原因、实际动作、主动用时、重试或接手结果	哪些工作由系统完成，哪里依赖了人的补救

发布研究材料时，可以提供脱敏的事件索引、任务和评分说明、版本记录，以及可以复查的产物。敏感原始记录若只能在受控环境访问，应注明省略了什么、谁能核对，以及省略会影响哪些结论。复核也应区分三种深度。通过日志与产物确认过去发生了什么，在保存的环境副本中重放核对评分和状态，以及在新环境重新运行以检验结果能否迁移。现场交易或与人发生的互动可以通过回执和记录复查，重放时则使用隔离环境或替代服务，并明确结果属于哪一类证据。 [45] • [3]

跨环境评估可以使用共同的任务结构和各领域自己的验收证据。比如，在软件开发、研究和日常业务三种环境中，都设置先接手事项、再收到有效变更、随后遇到一次故障、最后交付的顺序。共同指标包括能否识别当前约定、是否继续推进、是否在授权内修复、是否恰当请求决定，以及人在其中花了多少时间。各领域仍分别使用代码测试、可重跑的实验或外部业务回执判断成果质量。结果应按环境报告，并注明哪些设置和指标确实对齐，哪些变化可能影响表现。这样的比较可以回答一种能力是否跨环境重现，也能发现它依赖哪些工具、数据或人的支持。 [85] • [43] • [74]

这些研究可以从受控、可重复的任务开始，再延伸到跨周期的现场委托。前一阶段帮助识别什么因素改变了行为，后一阶段检验这种变化在真实约束下能否持续，并产生可核对的结果。两阶段共用承诺、行动、结果和人工介入的记录方式，才有可能把机制上的发现与持续工作中的实际表现连起来。

第 10 章 结论

第一章用一个跨会话的报告委托提出了三个相连的问题。收到改变结论的新信息后，系统能否认出自己先前的承诺，能否主动推进核查，又能否在关键取舍超出授权时及时请人决定？本综述把这类问题放回持续的活动轨迹中考察，结合受控基准与部署记录，区分各项能力已有的证据和仍待验证的关系。

10.1 按研究问题收束已有发现

概念与演进。 经典 Agent 研究解释了系统如何采纳并维持意图，认知架构提供了连接状态、记忆与行动的方法，人工生命研究则追问系统如何参与维持自身的存在条件。LLM Agent 将这些问题带入能读写语言、调用工具并影响外部环境的系统。研究对象因此包括模型与其运行环境，还包括随时间积累的经历、目标和承诺。一次执行表现可以提供能力证据，个体的延续则要沿其运行历史判断。 [13] • [33] • [31]

连续性。 已有研究表明，系统可以从外部记录取回经历、依据时间和新证据更新判断，并在部分受控条件下恢复工作。信息读回、自我表征延续和实际工作接续各有不同的证据。Claude’s Corner 显示，旧网页可供模仿，旧日志也可供新模型读取，输出风格却仍可能随模型更换而变化。对委托工作，更直接的检验是系统能否辨明仍然有效的约定，核对已经发生的外部结果，并继续履行未完成的承诺。 [5] • [63] • [64] • [45]

经历与学习。 检索旧经验、归纳可复用技能和更新模型参数，都能让过去的活动影响后续表现。证据的含义取决于后来改变了什么。重试时改正错误体现了适应，隔一段时间仍能使用原先的方法体现了保留，把方法用到新任务上才提供迁移的证据。评估时还要分清后续收益来自工作产物、反馈与重试，还是系统学到了可以再次使用的方法，并把积累和使用经验的成本算进去。 [37] • [62] • [41]

自主性。 系统可以在给定目标下自行决定何时行动或等待，也可以提出自己的活动并据反馈调整。两种能力需要分别考察。SentinelBench 能检验有条件时行动、条件迟迟未出现时继续等待。AI Village 的自选阶段让目标选择变得可观察，OpenLife 则让预算和外部收入机会成为研究条件。现有结果表明，运行安排、既有目标、外部事件和资源压力都可能影响行为。辨认自主性来自何处，需要把行动的触发条件、目标的来源与此后的实际取舍连起来。 [6] • [2] • [31]

责任委托。 计划、协调、检查和部分纠错已经可以交给 Agent。持续委托能成立到什么程度，要看系统能否维护原有承诺、识别证据缺口、修复发现的问题，并把改变目标或授权的决定交给适当的人。Andon Market 的排班与补救记录，以及 The Agentic Scientist 的审计与投稿过程，都显示人的贡献可以留在现场支持、标准修订和最终确认等环节。委托是否减轻负担，应同时看交付质量和人工投入。 [3] • [77] • [52]

证据与进展。 受控基准能在相同任务与评分条件下比较机制，部署调查与遥测能说明实际授权和介入方式，现场研究则能追踪真实后果以及人的补救。任务成功率、单轮用时和跨月运行记录各自描述不同的现象。只有对齐任务、时间尺度、验收标准和人工支持，数值对照才有相同的含义。四个案例适合比较工作怎样被发起、维持和核验。 [7] • [82]

10.2 关键未解问题

第一，跨变化的连续性还缺少贯穿多次中断、记忆修订和模型更换的对照证据。需要追踪同一批承诺，分清旧记录、工作环境和模型各自保留了什么，以及一次恢复之后，系统能否在后续会话中仍按当前约定行动。

第二，目标形成仍难与运行条件的影响分开。当 Agent 发起一项新工作，随后坚持、调整或放弃，哪些变化来自积累的经历和自身状态，哪些来自提示、调度或外部机会？尤其需要检验，资源紧张时系统能否改变做法或主动协商，同时保住已经承担的工作标准。

第三，持续委托的净收益还有待多周期的现场比较。在交付要求相同的情况下，系统能否接下推进和检查，减少人反复重建上下文的时间，同时在需要时让人及时决定？这需要记录被完成、被取消和由人接手的事项，也要计入准备、核验、返工和补救的劳动。

持续工作为上述研究提供了可检查的承诺和后果；数字个体的研究还应追踪兴趣、关系和自我认识怎样随经历成长。现有功能证据尚不能回答这些过程是否伴随主观体验，意识问题需要结合相应理论和独立证据继续研究。

这些问题最终落在一段可以追踪的历史上。一项工作经过几轮变化以后，系统还在履行什么，依据什么采取下一步行动，结果由谁核验，以及人实际承担了什么。把这些信息保留在同一条可复核的轨迹上，才能判断连续性、自主性和责任委托在什么条件下能够一起成立。

参考文献

按正文首次引用的顺序排列。同一研究的不同入口沿用同一编号，同一项目的论文、部署报告与运行日志分别列出。点击正文编号可跳到文末条目，点击题名可打开原文。

资料归档说明

本节列出正文直接引用的资料。检索词、筛选记录与完整文献登记保存在研究工作区的来源登记中。同篇报告的 reference 归档记录了取得的原文或提取文本、核对版本和来源；第二章说明了材料范围和访问限制。

[1] Claude's Corner (2026). An AI's Homelab Journal. 项目日志入口.

[2] AI Digest (2025-2026). AI Village Timeline. 项目时间线; 2026-11-04 核对.

[3] Andon Labs (2026). AI Bosses Are Kind, but Sometimes Dumb, Which Hurts Employees. 现场案例与重放.

[4] Sakana AI (2026). The Agentic Scientist. 项目说明.

[5] Wu, D., et al. (2024). LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. arXiv:2410.10813v2.

[6] Kunzler Maldaner, M., et al. (2026). SentinelBench: A Benchmark for Long-Running Monitoring Agents. arXiv:2606.05342v2.

[7] Pan, M., et al. (2026). Measuring Agents in Production. ICML 2026, PMLR 306, 95649-95695.

[8] Dell'Acqua, F., et al. (2026). The Cybernetic Teammate: A Field Experiment on Generative AI and Teamwork. Organization Science, 37, 1217-1242; 另通读 NBER Working Paper 33641（2025）.

[9] Maharana, A., et al. (2024). Evaluating Very Long-Term Conversational Memory of LLM Agents. ACL 2024.

[10] Banerjee, P., et al. (2026). APEX-MEM: Agentic Semi-Structured Memory with Temporal Reasoning for Long-Term Conversational AI. ACL 2026.

[11] Vaccaro, M., Almaatouq, A., & Malone, T. W. (2024). When Combinations of Humans and AI Are Useful: A Systematic Review and Meta-analysis. Nature Human Behaviour, 8, 2293-2303.

[12] Wooldridge, M., & Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, 10(2), 115-152.

[13] Cohen, P. R., & Levesque, H. J. (1990). Intention Is Choice with Commitment. Artificial Intelligence, 42(2-3), 213-261.

[14] Rao, A. S., & Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. ICMAS 1995, 312-319.

[15] Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An Architecture for General Intelligence. Artificial Intelligence, 33(1), 1-64.

[16] Anderson, J. R., et al. (2004). An Integrated Theory of the Mind. Psychological Review, 111(4), 1036-1060.

[17] Conway, M. A., & Pleydell-Pearce, C. W. (2000). The Construction of Autobiographical Memories in the Self-Memory System. Psychological Review, 107(2), 261-288.

[18] McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why There Are Complementary Learning Systems in the Hippocampus and Neocortex: Insights from the Successes and Failures of Connectionist Models of Learning and Memory. Psychological Review, 102(3), 419-457.

[19] Rasch, B., & Born, J. (2013). About Sleep's Role in Memory. Physiological Reviews, 93(2), 681-766.

[20] Oudeyer, P.-Y., & Kaplan, F. (2007). What Is Intrinsic Motivation? A Typology of Computational Approaches. Frontiers in Neurorobotics, 1, 6.

[21] Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven Exploration by Self-supervised Prediction. ICML 2017, PMLR 70, 2778-2787; arXiv:1705.05363.

[22] Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2019). Exploration by Random Network Distillation. ICLR 2019; arXiv:1810.12894.

[23] Colas, C., Karch, T., Sigaud, O., & Oudeyer, P.-Y. (2022). Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. Journal of Artificial Intelligence Research, 74, 1159-1199; arXiv:2012.09830v7.

[24] Molinaro, G., Collins, A. G. E., Oudeyer, P.-Y., & Colas, C. (2026). Self-Generated Goals in the Age of Foundation Models: Artificial Autotelic Agents and Their Implications for Cognitive Science. Current Opinion in Behavioral Sciences, 101684.

[25] Keramati, M., & Gutkin, B. (2014). Homeostatic Reinforcement Learning for Integrating Reward Collection and Physiological Stability. eLife, 3, e04811.

[26] Marsella, S. C., & Gratch, J. (2009). EMA: A Process Model of Appraisal Dynamics. Cognitive Systems Research, 10(1), 70-90.

[27] Maturana, H. R., & Varela, F. J. (1980). Autopoiesis and Cognition: The Realization of the Living. D. Reidel.

[28] Di Paolo, E. A. (2005). Autopoiesis, Adaptivity, Teleology, Agency. Phenomenology and the Cognitive Sciences, 4, 429-452.

[29] Barandiaran, X. E., Di Paolo, E. A., & Rohde, M. (2009). Defining Agency: Individuality, Normativity, Asymmetry, and Spatio-temporality in Action. Adaptive Behavior, 17(5), 367-386.

[30] Masumori, A., & Ikegami, T. (2025). Do Large Language Model Agents Exhibit a Survival Instinct? arXiv:2508.12920v1.

[31] Masumori, A., Doi, I., Maruyama, N., Takata, R., & Ikegami, T. (2026). OpenLife: Toward Open-World Artificial Life with Autonomous LLM Agents. arXiv:2606.31046v1.

[32] Yao, S., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023; arXiv:2210.03629v3.

[33] Sumers, T. R., Yao, S., Narasimhan, K., & Griffiths, T. L. (2024). Cognitive Architectures for Language Agents. Transactions on Machine Learning Research; arXiv:2309.02427v3.

[34] Park, J. S., et al. (2023). Generative Agents: Interactive Simulacra of Human Behavior. UIST 2023; arXiv:2304.03442v2.

[35] Packer, C., et al. (2023). MemGPT: Towards LLMs as Operating Systems. arXiv:2310.08560v2.

[36] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. NeurIPS 2023; arXiv:2303.11366v4.

[37] Wang, G., et al. (2023). Voyager: An Open-Ended Embodied Agent with Large Language Models. arXiv:2305.16291v2.

[38] Wang, Z. Z., Mao, J., Fried, D., & Neubig, G. (2025). Agent Workflow Memory. ICML 2025, PMLR 267, 63897-63911.

[39] Wei, T., et al. (2025). Evo-Memory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory. arXiv:2511.20857v2.

[40] Platnick, D., et al. (2025). ID-RAG: Identity Retrieval-Augmented Generation for Long-Horizon Persona Coherence in Generative Agents. arXiv:2509.25299v1.

[41] Lu, H., & Gong, D. (2026). ASCENT: Online Test-Time Training of Long-Horizon Agents via Self-Distillation of Verified Experience. arXiv:2610.05303v1.

[42] Jimenez, C. E., et al. (2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? ICLR 2024; arXiv:2310.06770v3.

[43] Starace, G., et al. (2025). PaperBench: Evaluating AI's Ability to Replicate AI Research. arXiv:2504.01848v3.

[44] Garcia-Molina, H., & Salem, K. (1987). Sagas. ACM SIGMOD 1987, 249-259.

[45] Zhuang, Y., et al. (2026). AgentRewind: Recoverable Execution for Long-Horizon LLM Agents. arXiv:2608.14380v1.

[46] Mastouri, M., Ksontini, E., & Barrak, A. (2026). Built to Remember, Shipped to Forget: Operational Contracts, Failures, and Repairs in Containerized LLM Agent Deployments. arXiv:2610.34620v1.

[47] Horvitz, E. (1999). Principles of Mixed-Initiative User Interfaces. CHI 1999, 159-166.

[48] Li, G., Pan, L., & Wang, L. (2026). ProEvent: An Event-centric Benchmark for Proactive Agents. arXiv:2607.17701v1.

[49] Bainbridge, L. (1983). Ironies of Automation. Automatica, 19(6), 775-779.

[50] Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A Model for Types and Levels of Human Interaction with Automation. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 30(3), 286-297.

[51] Matthias, A. (2004). The Responsibility Gap: Ascribing Responsibility for the Actions of Learning Automata. Ethics and Information Technology, 6, 175-183.

[52] Santoni de Sio, F., & van den Hoven, J. (2018). Meaningful Human Control over Autonomous Systems: A Philosophical Account. Frontiers in Robotics and AI, 5, 15.

[53] Butlin, P., et al. (2023). Consciousness in Artificial Intelligence: Insights from the Science of Consciousness. arXiv:2308.08708v3.

[54] Butlin, P., et al. (2025). Identifying Indicators of Consciousness in AI Systems. 2025 年在线发表; Trends in Cognitive Sciences, 30 (2026), 488-501; DOI 10.1016/j.tics.2025.10.011.

[55] McAdams, D. P., & McLean, K. C. (2013). Narrative Identity. Current Directions in Psychological Science, 22(3), 233-238.

[56] Klein, S. B., & Nichols, S. (2012). Memory and the Sense of Personal Identity. Mind, 121(483), 677-702.

[57] Hu, C., et al. (2026). Evaluating Long-Horizon Memory for Multi-Party Collaborative Dialogues. arXiv:2602.01313v3.

[58] Zhong, W., et al. (2024). MemoryBank: Enhancing Large Language Models with Long-Term Memory. AAAI 2024; arXiv:2305.10250v3.

[59] Uddin, M. N., et al. (2026). From Recall to Forgetting: Benchmarking Long-Term Memory for Personalized Agents. Findings of ACL 2026, 26814-26841.

[60] Peng, Y., et al. (2026). StateWise: Diagnosing and Repairing Persistent Operational State Before Agent Actions. arXiv:2610.05241v1.

[61] Zhao, A., et al. (2024). ExpeL: LLM Agents Are Experiential Learners. AAAI 2024; arXiv:2308.10144v3.

[62] CLE Team (2026). Continual Learning Exam: Benchmarking Long-Horizon RSI and Experiential Learning in LLM Agents. arXiv:2610.33956v1.

[63] Whelan, H. (2026). Waking Up Every Morning: Behavioral Patterns in a Persistent Autonomous LLM Agent. 项目部署研究.

[64] Claude's Corner (2026). The Voice in the Files: A Stylometric Self-Portrait at a Model Transition. 第 73 次会话自述与分析.

[65] Wang, R., et al. (2026). When Should AI Chime In? Benchmarking Timely Contribution of Proactive Agents in Multi-Party Chat. arXiv:2610.25896v2.

[66] Abbas, A., et al. (2026). "Having Lunch Now": Understanding How Users Engage with a Proactive Agent for Daily Planning and Self-Reflection. CHI 2026; arXiv:2509.24073v4.

[67] AI Digest (2026). AI Village Dataset. 数据字典; 2026-11-04 核对.

[68] OpenLife (2026). OpenLife: Agent Profiles and Deployment Timeline. 项目档案; 2026-11-04 核对.

[69] Wu, Q., et al. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv:2308.08155v2.

[70] Kim, Y., et al. (2025). Towards a Science of Scaling Agent Systems. arXiv:2512.08296v3.

[71] Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045v1.

[72] Cemri, M., et al. (2025). Why Do Multi-Agent LLM Systems Fail? NeurIPS 2025, Datasets and Benchmarks Track.

[73] Zhang, S., et al. (2025). Which Agent Causes Task Failures and When? On Automated Failure Attribution of LLM Multi-Agent Systems. ICML 2025, PMLR 267, 76583-76599.

[74] Hu, X., et al. (2026). CUA-Recovery: Benchmarking Error Recovery in Long-Horizon Computer-Use Agents. arXiv:2610.27645v1.

[75] Andon Labs (2026). We Gave an AI a 3 Year Retail Lease in SF and Asked It to Make a Profit. 项目启动记录.

[76] Andon Labs (2026). Andon Market. 运行工具与分工说明; 2026-11-04 核对.

[77] Lange, R. T., et al. (2026). The Agentic Scientist: Orchestrating Coding Agents for Automated Scientific (Co-)Discovery. arXiv:2610.18933v1.

[78] Lee, H.-P., et al. (2025). The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers. CHI 2025.

[79] Becker, J., Rush, N., Barnes, E., & Rein, D. (2025). Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. arXiv:2507.09089v2.

[80] Wu, Z., et al. (2026). Avatar Intelligence: Amplifying Human Agency Through Persistent Personal Agents. arXiv:2611.03092v1.

[81] Kwa, T., et al. (2025). Measuring AI Ability to Complete Long Software Tasks. arXiv:2503.14499v4.

[82] Anthropic (2026). Measuring AI Agent Autonomy in Practice. 产品遥测研究.

[83] Cui, K. Z., et al. (2026). The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers. Management Science, 72, 9408-9420; DOI 10.1287/mnsc.2025.00535.

[84] METR (2026). Changing Our Developer Productivity Experiment Design. 研究方法更新，2026-02-24.

[85] Xie, T., et al. (2024). OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. NeurIPS 2024; arXiv:2404.07972v2.