

NavHarness: 迈向终身具身导航

Xunyi Zhao^{1,2,*} Jian Zhou^{1,*} Sihao Lin^{1,2} Gengze Zhou¹ Zerui Li¹
Xinyu Yan^{1,2} Jiajun Liu^{2,3} Anton van den Hengel^{1,2} Qi Wu^{1,2,†}

¹ 阿德莱德大学 ² AIML 负责任人工智能研究中心 ³ CSIRO Data61

* 同等贡献, † 通讯作者

摘要

如今, 前沿模型已能借助简单工具进行多轮多模态推理, 在单个具身导航任务上取得良好表现。然而, 在连续任务中, 智能体还必须依赖不断演化的地图和早期搜索记录, 而二者都可能不完整, 或与新观测相冲突。本文提出 **NavHarness**, 一种面向终身导航、无需训练的具身执行框架 (Embodied Harness), 将记忆处理纳入导航循环。在导航过程中, 其多轮自主会话会利用地图、任务记录和房屋知识, 并根据观测核验这些信息、记录修正结果以指导行动。NavHarness 在为新任务或恢复尝试开启全新对话时保留这些经验, 同时通过结果验证和运行结束总结支持后续复用。在 GOAT-Bench 上, 与仅使用上下文的独立会话相比, NavHarness 搭配 Astra 时将 s-SR 提高了 18.6 个百分点, 搭配 Opus 5 时提高了 22.6 个百分点。使用 SLAM 估计的位姿时, 搭配 GPT-6 Astra 的 NavHarness 在 GOAT-Bench 上取得 83.7 的 s-SR 和 36.9 的 e-SR, 在 IR2R-CE 上取得 85.9 的 s-SR, 达到最先进的任务成功率。为理解这些提升, 本文考察了经验如何在会话之间传递, 发现结构化恢复交接记录优于等长总结。在跨房屋的扩展部署中, 与仅保留地图和任务记录相比, 整合进一步改善了导航; 案例研究表明, 智能体会利用早期经验解释新目标、调查未决问题并恢复失败的搜索。本文认为, 除提升单任务能力外, 终身导航的进展还取决于连续推理会话如何利用先前经验。

日期: 2026 年 9 月 28 日

项目主页: <https://billzhao1030.github.io>

GitHub: <https://github.com/billzhao1030/NavHarness>

1 引言

设想一个全天在住宅中导航的家用机器人: 它接连执行请求, 重访熟悉的房间, 并穿行于尚未探索的区域。多轮多模态推理的进展让这一愿景更加可行: 前沿模型无需针对导航进行专门训练, 仅借助基础编码执行框架及简单的观测和移动工具, 便可在单个任务内有效导航 (Zhou et al., 2026)。对于持续工作的机器人, 每段行程还会揭示更多住宅信息, 并留下可帮助后续请求的经验。要将这种能力扩展至终身导航, 机器人就必须在决定前往何处、接下来检查何时利用这些经验。

在连续的家居导航中, 每个任务都从上一任务的终点开始, 因此新请求到来时, 机器人已经探索了住宅的一部分。重启对话不会改变机器人的位置, 新的推理会话因而需要此前积累的地图和搜索证据。然而, 某条记录称在某个房间未找到物体, 并不能证明所有相关区域都经过检查; 保存的描述也可能与当前视野冲突。为决定下一步搜索位置, 导航器必须结合当前观测解释这些记录, 并在证据发生变化时修订记录。

长时程导航已朝这一目标取得显著进展, 包括跨任务保留场景知识 (Zhao et al., 2024)、检索过去的观测以支持探索 (Yang et al., 2025; Niu et al., 2026; Lu et al., 2026), 以及学习查询累积经验的策略 (Wang et al., 2026c)。在此基础上, 本文研究通用推理会话如何在导航时综合利用这些不同的信息来源。例如, 房屋笔记可能提示某个房间, 地图可以给出前往该房间的路线, 而失败搜索记录则会指出某个被忽略的角落。到达房间后, 机器人可将这些线索与眼前观测进行比较, 据此决定是继续在此搜索、查阅另一条记录, 还是修正笔记。这样, 记忆检索与修订便成为与环境持续交互的一部分。

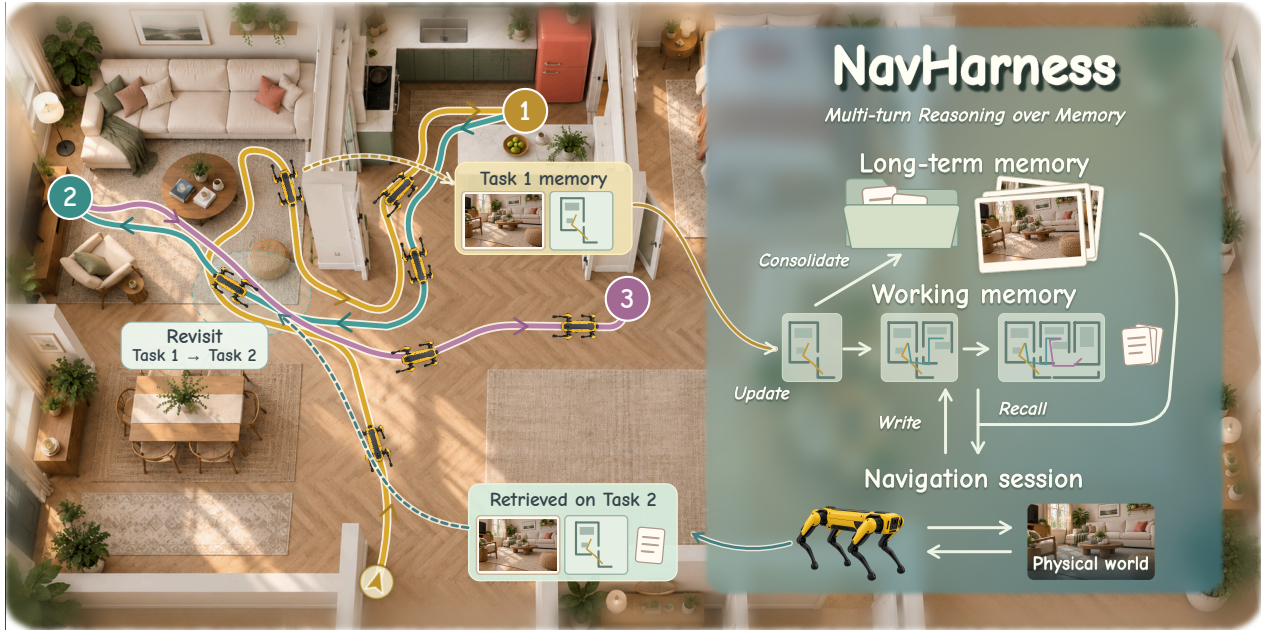


Figure 1 NavHarness 通过对工作记忆与房屋知识进行多轮多模态推理，编排长时程具身导航，并为后续任务保留经验。

多轮自主会话是一个切实可行的起点，因为记忆智能体和编码智能体已经能够使用工具检查外部记录，并依据发现进一步查询 (Packer et al., 2023; Rajasekaran et al., 2025; Young, 2025)。在导航中，这种交互使会话能够读取搜索记录，将其与当前视野比较，并在选择下一步行动前查阅地图，而无须训练单独的内存策略。将这些记录置于对话之外，也使新会话能够继承有关已检查区域、证据支持的排除结论及未决搜索的信息。后续搜索会依赖这些记录，因此结果检查有助于避免错误的完成声明变成被接受的经验。

本文提出 **NavHarness**，一种面向终身导航、以记忆处理为核心且无需额外模型训练的具身执行框架。每个多轮自主会话基于当前观测、空间记忆 (Spatial Memory)、任务记忆及更长期的房屋知识进行推理；外层循环则在新任务或恢复尝试开始全新对话时保留这些记忆。该循环还会检查任务结果，并将生成的记录整合到房屋笔记中，供后续运行使用。在 GOAT-Bench 上，与同一模型为每项任务启用不含外部记忆的全新纯上下文会话相比，NavHarness 搭配 Astra 时将 s-SR 提高了 18.6 个百分点，搭配 Opus 5 时提高了 22.6 个百分点 (第 4.2 节)。使用 SLAM 估计位姿而非模拟器真实位姿时，搭配 GPT-6 Astra 的 NavHarness 在 GOAT-Bench 上达到 83.7% 的 s-SR 和 36.9% 的 e-SR，在 IR2R-CE 上达到 85.9% 的 s-SR，超过此前最先进结果。扩展部署展示了保留的经验如何改变搜索方向、促使智能体通过观测消除不确定性，并让失败尝试为后续决策提供信息。

本文的贡献总结如下：

1. 本文表明，除保留经验之外，使用经验的方式同样重要。结构化交接记录优于等长总结。NavHarness 将这种证据传递机制纳入导航循环，使全新推理会话能够从机器人当前状态出发，利用早期搜索证据继续导航。
2. 本文提升了全部四种骨干模型的性能，并使用 SLAM 估计位姿而非模拟器真实位姿，在 GOAT-Bench 和 IR2R-CE 上取得最先进表现；与采用相同骨干模型的独立会话相比，任务成功率和路径效率均有所提高。
3. 连续部署表明，与仅保留地图和任务记录相比，整合能够进一步改善导航。案例研究揭示了智能体如何复用失败搜索并修订早期记录，为未来长时程具身智能体的设计提供依据。

2 相关工作

长时程与终身导航。IR2R-CE 和 GOAT-Bench 在共享环境中评估连续目标，累积经验能够改善后续导航 (Krantz et al., 2023; Khanna et al., 2024)。OVER-NAV、3D-Mem、SSMG-Nav 和 GSMem 分别通过结构化对象、视觉快照、语义图及可渲染记忆保留空间知识 (Zhao et al., 2024; Yang et al., 2025; Niu et al., 2026; Lu et al.,

2026), 帮助智能体将过去的观测与可重访的位置关联起来。在场景表征之外, SeqWalker 使用分层规划与轨迹修正 (Han et al., 2026), MemoryExplorer 通过单轮工具调用训练主动检索 (Wang et al., 2026c), Uni-Walker 和 AllDayNav 则探索持续策略学习 (Wang et al., 2026d; Yin et al., 2026)。本文具有相同目标, 但关注通用多轮会话如何在无须针对导航进行专门训练的情况下, 查阅并修订空间记忆、任务记录和房屋知识。

具身执行框架与自主控制。MIP 表明, 多轮多模态推理可在基础编码执行框架中通过观测和移动工具控制导航 (Zhou et al., 2026)。导航接口与空间工具扩展了这种方法 (Li et al., 2026d; Deng et al., 2026; Lei et al., 2026), 具身执行框架则将推理与机器人控制器及可复用技能相连接 (Zhang et al., 2026; Chen et al., 2026d; Wang et al., 2026b; Elmaaroufi et al., 2026)。持久化经验还支持技能与执行框架的演化 (Wang et al., 2024; Ding et al., 2026; Wang et al., 2026a)。HarnessVLN 使用事件记忆和时空图验证导航方案 (Chen et al., 2026b)。RoboHarness 使用执行记忆选择异构策略, 并为策略交接准备物理状态 (Huang et al., 2026)。与之不同, NavHarness 研究推理会话边界之间必须传递哪些信息。全新对话从机器人当前的物理状态开始, 因此需要了解哪些位置已检查、这些观测排除了什么, 以及哪些搜索尚未完成。本文检验检索、交接和整合这些证据如何支持后续导航。

长时程智能体中的记忆处理。外部记忆使智能体能够在上下文窗口之外保留经验, 而无须将每段新经验编码进模型参数。Generative Agents 和 A-MEM 将观测转化为可检索、可修订的记录 (Park et al., 2023; Xu et al., 2025), Reflexion 和 ACE 则保留早期尝试中的经验教训 (Shinn et al., 2023; Zhang et al., 2025b)。为在有限上下文窗口内使用不断增长的记忆, 智能体会采用压缩、选择性淘汰和由智能体主导的上下文管理 (Kang et al., 2026; Semenov & Dorofeev, 2026; Hao et al., 2026; Li et al., 2026b)。MemGPT 允许模型在不同记忆层级间移动信息, CoALA 则将记忆访问纳入智能体的决策过程 (Packer et al., 2023; Summers et al., 2024)。在编码执行框架中, 多轮工具交互使智能体能够读取和更新持久文件, 在会话之间传递进展 (Yang et al., 2024; Wang et al., 2025; Rajasekaran et al., 2025; Young, 2025; Rajasekaran, 2026)。NavHarness 将这些实践用于物理搜索, 而核验记录在此场景中可能需要重访相应位置。固定检索和普通总结对照实验用于检验智能体自主访问及结构化搜索交接记录的价值。

3 NavHarness

术语	含义与关系
运行 (部署)	在一所房屋中连续执行的 K 个任务。
任务	对目标 g_k 的追寻, 可能包含共享同一任务预算的多次尝试。
尝试	针对当前目标的一次连续搜索, 由一个导航会话支持。
会话 (导航)	作为一次尝试推理核心的多轮多模态对话。
步段	会话内一组有界的模型轮次与工具调用。

任务设置。与单回合导航评估 (Anderson et al., 2018b) 不同, 本文研究具有可复用经验的连续导航目标。每个目标由语言、物体类别或图像指定。在一个连续序列中, 每项任务均从上一任务的终点开始。恢复 (Recovery) 会替换对话, 同时保留物理状态、外部记忆和剩余预算, 以便对同一目标进行另一次尝试。任务闭合 (Task Closure) 会在进入下一个目标前记录结果, 预算耗尽时也不例外。运行结束时, 整合 (Consolidation) 会总结任务记录, 供未来运行使用。

外部编排器 (Orchestrator) 管理会话边界, 处理恢复和完成请求, 并保留记忆。它以步段推进会话而不清空对话, 并在步段之间跟踪任务预算和机器人状态。图 1 和图 2 分别展示记忆复用与执行流程。附录 A 给出了完整执行循环 (算法 1)、六组件规范 (Meng et al., 2026) 及详细转换规则。

3.1 利用保留的经验启动任务

收到新目标后, 编排器设定模型轮次和移动步数预算, 并开启一个可访问持久工作记忆 (Working Memory) 的全新编码智能体对话, 要求其在移动前读取引用的文件。如果相关文件可用, 这些引用会指向上一任务的交接记录、运行台账和房屋笔记。这些内容通过文件读取进入对话, 而不是完整插入初始提示; 随着搜索推进, 会话还可查阅其他记录。交接记录 (Handover) 为后续会话保留搜索经验。任务闭合时, 任务交接记录总结目标、已搜索位置、结果证据和未完成搜索。恢复期间, 恢复记录则将证据和剩余选项传递给针对同一目标的新尝试 (第 3.3 节)。台账 (Ledger) 是一个紧凑索引, 每项已闭合任务对应一个条目, 其中列出目标、记录的状态和具名位置。它帮助后续会话找到相关任务, 并通过文件工具打开详细记录, 无须继承早期对话。

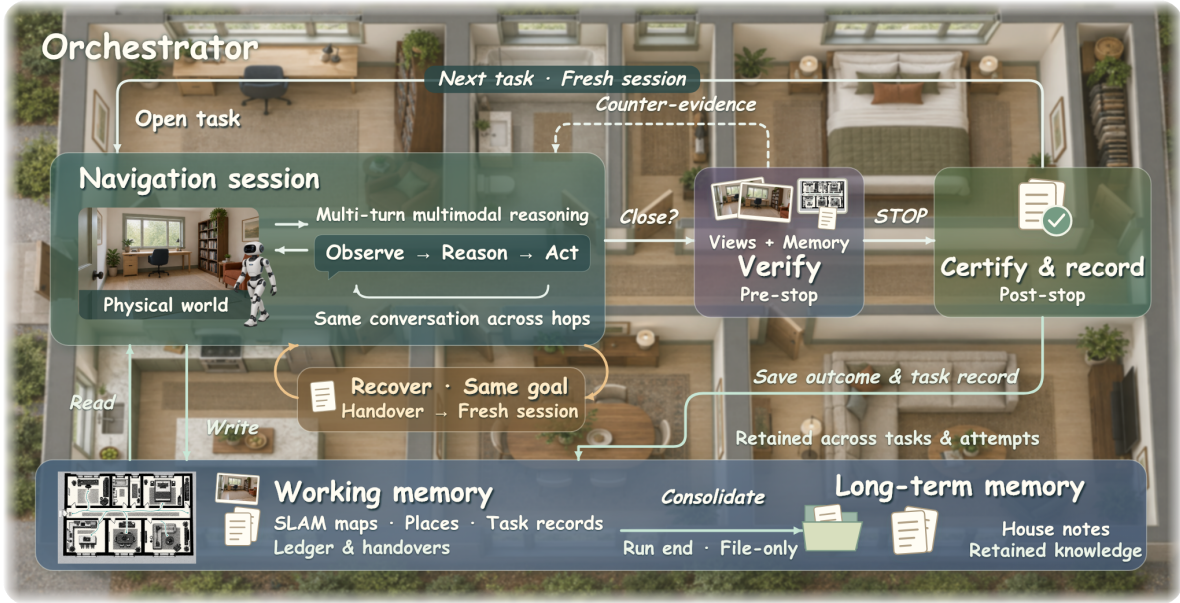


Figure 2 NavHarness 执行流程。编排器管理导航会话和持久记忆。Verify 和 Certify 分别表示停止前验证与停止后认证。

导航会话自行写入任务交接记录和恢复记录，编排器则写入台账及裁决文件。导航会话可以读取早期任务记录，但不能覆盖它们。

导航会话访问三个记忆层级。活动上下文 (*Active Context*) 包含当前尝试的目标、推理、观测及检索到的信息。工作记忆位于对话之外，并在任务和尝试之间持久保留；它将任务记录与空间记忆配对，后者包括利用 SLAM 估计位姿构建的占据栅格 (*Occupancy Grid*)、具名位置及其照片，以及房间和楼层之间的连通关系。组成日志的任务记录包括台账、任务交接记录、恢复记录和完成检查结果 (第 3.4 节)。长期记忆 (*Long-Term Memory*) 由四个文件组成，统称为房屋笔记 (*House Notes*)，包括索引、房屋概览、房间笔记和导航技能。其内容涵盖房间连通关系、地标、实用路线、失败搜索、被证据排除的位置及未决问题。导航会话读取这些文件以指导搜索，但只有整合过程会更新它们 (第 3.5 节)。保存的地图和视图独立于这些文本笔记持久化，以便在后续运行中复用 (附录 A)。因此，这些层级描述的是记忆的使用方式及其可用时长，并非三个互不相交的存储。当前运行中由多个任务共享的地图在保存以供后续运行使用时，会成为保留的房屋记忆的一部分，而无须转换为文本总结。

3.2 通过查阅并更新记忆进行导航

从上述上下文出发，导航会话交替进行观测、推理、行动和进一步的记忆访问 (Zhou et al., 2026)。房屋笔记可以提示某个房间，任务记录可以说明哪些地方已经搜索过，地图查询则可以展示如何到达该房间。工具响应会将检索到的信息带入对话，以便与新观测比较，帮助模型决定是采信早期记录还是继续查询。遵循既有工作，本文采用标准导航动作空间：向前移动 (0.25 m)、左转/右转 (15°) 及 STOP。

到达记忆中的位置还需要定位机器人，并在已探索空间中规划路线。ORB-SLAM3 (Campos et al., 2021) 根据彩色与深度 (RGB-D) 观测估计机器人的位置和朝向 (位姿)。建图器使用这些估计结果为每层楼构建占据栅格，以区分自由、占用和未探索空间。机器人移动时，只要位姿跟踪可用，建图器就会更新栅格；若跟踪丢失，则暂停更新。导航会话查询这张不断扩展的地图，以检查已探索空间并预览路线，还可用保存的视图标记具名位置，供后续参考 (附录 A.2)。占据栅格通过估计位姿投影深度观测构建，而非直接使用跟踪器的点云。路线查询在已知自由栅格上运行 A^* ，并返回路径及其长度供会话判断。具名标记将栅格中的位置与保存的视图关联起来，使后续会话能够比较记忆中的位置与当前所见内容。

闭合和恢复工具会将请求传递给编排器，由编排器在步段返回时处理。编排器会检查剩余预算和跟踪状态。如果环境已结束任务，编排器便记录结果；否则，先处理待定闭合请求，再考虑恢复。若二者都不要求状态转换，则在保留完整上下文的情况下恢复同一对话。编码后端负责管理该对话中的消息记录，而 NavHarness 在任务边界和恢复边界开启全新会话。将消息记录管理与这些边界分离，使不同推理核心能够使用同一套记忆规则 (附录 A.3)。NavHarness 不会在运行中的会话内额外采用图像淘汰窗口或消息记录总结策略。任务的

轮次预算和移动预算涵盖其全部步骤与尝试，因此替换对话不会获得新的搜索预算。

3.3 通过全新对话进行恢复

当会话反复执行无效方案时，恢复机制允许全新对话利用保留的证据重新考虑搜索。恢复由会话发起请求，或在尝试达到设定的轮次阈值时开始。编排器会检查探索进度、跟踪状态及先前尝试，再决定是否批准重启，以及是否还需重新评估位置。替换会话前，即将结束的会话会写入一条恢复记录，作为同一目标下一次尝试的交接记录。移动被禁用时，该记录会记载已搜索位置、有证据支持的排除结论、可靠地标、尚未尝试的选项及不确定性。区分已搜索与已排除非常重要，因为到访一个房间并不能证明看到了其中所有相关物体。恢复记录会注明支持每项排除结论的观测，使下一次尝试能够区分反对某一位置的证据与该处尚未完成的搜索。全新会话读取此记录，并从当前位置继续执行同一目标；它会保留地图、任务记录、房屋笔记及剩余预算，但不继承旧对话。

恢复范围规定新尝试应重新考虑哪些内容，可以从搜索方案扩展至位置、路线、楼层或对房屋的理解，而不删除已存储的记忆。若需要重新评估位置，独立的唤醒会话会将周围视图与保存的位置照片进行比较，并向新的导航会话提供一段关于可能所处位置的简短描述。在没有保存照片时，导航会话依据给定视图推断自身位置。这种视觉识别有助于确定搜索方位，而 SLAM 负责估计机器人的度量位姿（附录 A.5）。恢复范围改变的是需要重新考虑的信念，而非机器人的位置或已存储观测。与恢复不同，任务序列之间预设的位置迁移会在下一任务开始前移动机器人。此时，新的导航会话会获得周围视图，并利用保留的记忆识别当前环境。

3.4 检查完成情况并传递任务记录

当导航会话决定停止搜索时，它会写入关于搜索过程、结果及剩余不确定性的任务交接记录，然后请求闭合并给出自身的完成情况判断。后续任务可能依赖这份记录，因此一个独立的评判会话会利用目标、停止位置的四幅视图、采样的早期帧、地图及执行记录进行验证。评判会话无法使用工具，也无法访问导航对话或导航器的自我评估，只会返回有证据支持的完成 (*complete*)、未完成 (*incomplete*) 或未知 (*unknown*) 裁决。视觉证据最多包括十二帧，其中优先采用停止位置的四幅视图，其余从早期观测中采样。

对于停止前验证 (*Pre-Stop Verification*)，编排器会在发出 STOP 前，将评判会话的独立裁决与导航器的完成声明比较。如果首次接受检查的请求得到的明确裁决与声明相矛盾，编排器就会向当前导航会话返回证据，使其有机会继续搜索。未知裁决不会阻止闭合，重复请求也会获准执行。这项一次性检查属于任务本身，不会因恢复而重置，从而避免重复否决无限期推迟闭合。停止后认证 (*Post-Stop Certification*) 会记录评判会话的最终评估，但不改变任务得分。如果动作步数未发生变化，则复用早期裁决；否则检查最终证据（附录 A.6）。

编排器会保存裁决，在台账中追加条目，并为下一任务的对话保留空间记忆；任务因预算耗尽而结束时亦是如此。完成声明与裁决彼此分离，因为完成检查并不能验证交接记录中的每项空间断言。

3.5 为下一次运行整合经验

为避免后续会话检查过去的每一条任务记录，独立的整合会话会在运行结束时读取日志和现有房屋笔记，随后重写索引、房屋概览、房间笔记及导航技能。仅当某项技能得到日志支持时才将其加入。整合会话只使用文件工具，在区分已检查结果与导航器声明并保留源记录的同时，整合经验和修正内容（附录 A.4）。其写入权限仅限于四个房屋笔记文件，因此整合既不会改变地图几何，也不会覆盖底层任务记录。在写入这份总结之前，地图、位置记录和任务经验就已经可以持久保留。

在同一房屋的后续运行中，导航会话会结合保存的地图与视图读取这些笔记，将过去经验与当前环境联系起来；在任务序列之间进行预设位置迁移后同样如此。本文在第 4.4 节评估这种跨运行复用，而第 4.2 节和第 4.3 节中的基准运行均从空记忆开始。

4 实验

4.1 实验设置

我们在 GOAT-Bench (Khanna et al., 2024) 和 IR2R-CE (Krantz et al., 2023) 的验证集未见划分上评估 NavHarness。在 GOAT-Bench 上，评估覆盖 36 个场景、360 个回合中的全部 2,669 个子任务。推理核心包括 Qwen3.8-27B (Qwen Team, 2026)、GPT-4o (OpenAI, 2024)、Opus 5 (Anthropic, 2026) 和 GPT-6 Astra (OpenAI, 2026)，均配备

相同的导航与记忆工具。同一模型内的比较使用相同评测任务和任务预算。本文报告单任务成功率 (s-SR) 和每项任务均成功的序列比例 (e-SR) (Niu et al., 2026), 并报告衡量路径效率的 SPL (Anderson et al., 2018a), 以及 IR2R-CE 中衡量轨迹保真度的 t-nDTW (Krantz et al., 2023; Ilharco et al., 2019)。附录 B 给出后端配置与详细评测设置。在 GOAT-Bench 回合中, 记忆从一个子任务延续至下一子任务; 在 IR2R-CE 游程中, 记忆在任务间延续。每个新回合或游程均以空地图、空任务记录和空房屋笔记开始, 不继承上一次运行的记忆。

4.2 与最先进方法的比较

Table 1 GOAT-Bench 验证集未见划分上与最先进方法的比较。GT 位姿: 模拟器真实位姿。[†]: 在 278 个子任务上评估。本文结果: 三个随机种子的均值 $\pm$ 标准差。

方法	基础模型	类型	GT 位姿	s-SR	SPL	e-SR
MemoryExplorer [†] (Wang et al., 2026c)	Qwen2.5-VL-7B	训练式	✓	46.4	28.0	–
SSMG-Nav (Niu et al., 2026)	Qwen-VL-Plus	零样本	✓	46.5	34.1	8.6
EvoMemNav (Ge et al., 2026)	Qwen3-VL-8B	零样本	✓	59.6	38.9	–
ReEXplore [†] (Zhang et al., 2025a)	GPT-4o	零样本	✓	59.8	42.5	–
AstraNav-Memory (Ren et al., 2025)	Qwen2.5-VL-3B	训练式	✓	62.7	56.9	–
STEGNav (Chen et al., 2026c)	GPT-5.4-mini	零样本	✓	66.3	39.7	–
GSMem (Lu et al., 2026)	GPT-4o	零样本	✓	67.2	46.9	–
3D-Mem [†] (Yang et al., 2025)	GPT-4o	零样本	✓	69.1	48.9	–
MetaNav (Li et al., 2026c)	GPT-4o	零样本	✓	71.4	51.8	–
HGR [†] (Chen et al., 2026a)	GPT-4o	零样本	✓	72.4	56.2	–
ObsGraph [†] (Lee et al., 2026)	GPT-4o	零样本	✓	72.7	51.5	–
HIMM [†] (Li et al., 2026a)	GPT-4o	零样本	✓	72.8	56.1	–
本文方法 (相同骨干模型比较)						
独立会话				43.5 $\pm$ 0.6	30.2 $\pm$ 0.5	6.4 $\pm$ 0.6
NavHarness (子集) [†]	GPT-4o	自主式	×	77.6 $\pm$ 1.0	56.4 $\pm$ 0.8	25.0 $\pm$ 2.8
NavHarness				78.3 $\pm$ 0.5	57.1 $\pm$ 0.4	26.9 $\pm$ 0.8
独立会话				41.4 $\pm$ 0.6	27.5 $\pm$ 0.5	3.1 $\pm$ 0.5
NavHarness	Qwen3.8-27B	自主式	×	71.7 $\pm$ 0.5	48.2 $\pm$ 0.4	14.4 $\pm$ 0.7
独立会话				58.9 $\pm$ 0.9	34.9 $\pm$ 0.8	5.8 $\pm$ 0.6
NavHarness	Opus 5	自主式	×	81.5 $\pm$ 0.7	55.0 $\pm$ 0.6	28.6 $\pm$ 0.8
独立会话				65.1 $\pm$ 0.6	40.1 $\pm$ 0.5	13.6 $\pm$ 0.7
NavHarness	GPT-6 Astra	自主式	×	83.7 $\pm$ 0.4	62.3 $\pm$ 0.3	36.9 $\pm$ 0.9

固定各推理模型时, 相较独立会话, NavHarness 在 GOAT-Bench 上将 Qwen3.8-27B、GPT-4o、Opus 5 和 GPT-6 Astra 的 s-SR 分别提高 30.3、34.8、22.6 和 18.6 个百分点 (表 1)。这些 MIP 式对照 (Zhou et al., 2026) 仅使用当前任务上下文, 因此增益反映完整具身执行框架的作用, 包括建图、恢复和完成检查。第 4.3 节和第 4.4 节进一步分离记忆的贡献。Astra 的 SPL 也从 40.1 提升至 62.3; 其在 IR2R-CE 上的 s-SR 增益为 17.5 个百分点, 接近其在 GOAT-Bench 上的 18.6 个百分点。

采用相同 GPT-4o 骨干模型和 278 子任务协议时, NavHarness 的 s-SR 达到 77.6, 且无需模拟器位姿, 超过 HIMM (72.8) 和 3D-Mem (69.1) (Li et al., 2026a; Yang et al., 2025)。采用 Astra 时, NavHarness 在 GOAT-Bench 上达到 83.7% s-SR 和 62.3 SPL, 分别比表 1 中此前最优结果高 10.9 和 5.4 个百分点 (Li et al., 2026a; Ren et al., 2025)。在 IR2R-CE 上 (表 2), 其达到 85.9% s-SR 和 76.1 SPL, 而 SeqWalker 分别为 36% 和 34 (Han et al., 2026)。开放的 27B Qwen 执行器在 GOAT-Bench 上也达到 71.7% s-SR, 较已发表的 3D-Mem 结果高 2.6 个百分点 (Yang et al., 2025)。NavHarness 无需模拟器位姿、全局导航网格、辅助感知模型或导航专用学习策略即可取得这些结果。

Table 2 IR2R-CE 验证集未见划分。GT 位姿：模拟器位姿。本文结果：三个随机种子的均值 $\pm$ 标准差。

方法	类型	GT 位姿	s-SR	SPL	e-SR	t-nDTW
CMA (Krantz et al., 2023)	训练式	×	19	18	—	38
MAP-CMA (Krantz et al., 2023)	训练式	✓	35	32	—	47
ETPNav (Han et al., 2026)	训练式	✓	28	27	—	41
HNR (Han et al., 2026)	训练式	✓	30	28	—	44
OVER-NAV (Zhao et al., 2024)	混合式	✓	35	33	—	50
SeqWalker (Han et al., 2026)	训练式	✓	36	34	—	52
本文方法（相同骨干模型比较）						
独立会话（Qwen3.8-27B）	自主式	×	29.0 $\pm$ 0.9	21.0 $\pm$ 1.1	0.0 $\pm$ 0.0	36.9 $\pm$ 1.2
NavHarness（Qwen3.8-27B）	自主式	×	45.1 $\pm$ 1.1	32.0 $\pm$ 0.9	0.0 $\pm$ 0.0	43.2 $\pm$ 1.2
独立会话（GPT-4o）	自主式	×	37.0 $\pm$ 1.3	29.0 $\pm$ 1.2	2.8 $\pm$ 2.8	43.1 $\pm$ 0.9
NavHarness（GPT-4o）	自主式	×	57.8 $\pm$ 1.2	45.2 $\pm$ 1.0	13.9 $\pm$ 2.8	52.9 $\pm$ 1.1
独立会话（Opus 5）	自主式	×	59.3 $\pm$ 1.1	32.8 $\pm$ 1.2	2.8 $\pm$ 0.0	45.1 $\pm$ 1.3
NavHarness（Opus 5）	自主式	×	78.2 $\pm$ 0.9	56.9 $\pm$ 1.3	16.7 $\pm$ 2.8	58.1 $\pm$ 1.0
独立会话（GPT-6 Astra）	自主式	×	68.4 $\pm$ 1.4	53.6 $\pm$ 0.8	9.3 $\pm$ 1.6	56.8 $\pm$ 1.2
NavHarness（GPT-6 Astra）	自主式	×	85.9 $\pm$ 0.8	76.1 $\pm$ 0.9	27.8 $\pm$ 2.8	64.2 $\pm$ 1.1

对终身导航而言，这些改进必须贯穿完整目标序列。采用 Astra 时，GOAT-Bench 的 e-SR 从独立会话的 13.6% 升至 NavHarness 的 36.9%，意味着超过三分之一的游程能完成所有子任务。表 1 报告的此前最高 e-SR 为 SSMG-Nav 的 8.6% (Niu et al., 2026)。在 IR2R-CE 上，Astra 的 e-SR 从 9.3% 升至 27.8%。

4.3 哪些机制使导航经验跨任务、跨尝试延续？

我们使用 Opus 5，在匹配任务预算的条件下评估 GOAT-Bench 验证集未见划分中的全部 2,669 个子任务。表 3 比较会话策略（A）、跨任务记忆（B）、恢复交接记录（C）和完成检查（D）。恢复与原任务共享预算（附录 B.6 和附录 B.4）。

Table 3 Opus 5 在完整 GOAT-Bench 验证集未见划分上的比较。结果为三个随机种子的均值 $\pm$ 标准差（%）。 Δ s-SR 表示相较完整系统的配对变化。

配置 / 干预	s-SR	SPL	Δ s-SR	95% 置信区间
NavHarness（完整参考系统）	81.5 $\pm$ 0.7	55.0 $\pm$ 0.6	—	—
A. 会话策略				
独立会话	58.9 $\pm$ 0.9	34.9 $\pm$ 0.8	-22.6	[-26.3, -19.0]
单一会话	54.9 $\pm$ 1.1	32.8 $\pm$ 0.9	-26.6	[-30.5, -22.8]
独立会话 + 停止前验证	62.4 $\pm$ 0.8	36.5 $\pm$ 0.7	-19.1	[-22.4, -16.4]
B. 跨任务记忆				
每项任务后清空地图（保留记录）	68.9 $\pm$ 0.8	39.8 $\pm$ 0.7	-12.6	[-15.6, -9.7]
隐藏早期任务记录（保留地图）	71.8 $\pm$ 0.7	41.8 $\pm$ 0.6	-9.7	[-12.5, -7.0]
无跨任务记忆	66.8 $\pm$ 0.9	37.4 $\pm$ 0.7	-14.7	[-16.9, -12.4]
C. 恢复交接记录				
无恢复（任务预算相同）	71.1 $\pm$ 0.9	46.1 $\pm$ 0.8	-10.4	[-13.2, -7.7]
恢复 + 空交接记录	72.9 $\pm$ 0.8	44.7 $\pm$ 0.7	-8.6	[-11.3, -6.0]
恢复 + 其他任务的交接记录	68.5 $\pm$ 1.0	44.8 $\pm$ 0.8	-13.0	[-16.1, -10.0]
恢复 + 等长摘要	73.2 $\pm$ 0.7	47.0 $\pm$ 0.6	-8.3	[-11.0, -5.7]
D. 完成检查				
无停止前验证	76.1 $\pm$ 0.8	47.0 $\pm$ 0.7	-5.4	[-7.8, -3.1]
无停止后认证	79.0 $\pm$ 0.6	48.2 $\pm$ 0.6	-2.5	[-4.5, -0.6]
仅一个停止视图而非四个	78.6 $\pm$ 0.7	47.9 $\pm$ 0.6	-2.9	[-5.0, -0.9]

A. 会话策略。单一会话使对话跨任务延续，仅在上下文窗口填满时使用后端压缩，但其表现仍弱于独立会话（54.9% 对 58.9%）。独立会话在每项任务开始时仅保留当前对话，不使用外部记忆。较长的对话还会保留过时目标和观测，可能与当前任务相关信息形成竞争。NavHarness 为每项任务启动全新对话，同时通过外部记忆保留先前经验。即使为独立会话加入停止前验证，与 NavHarness 仍有 19.1 个百分点的差距。其余比较考察该差距背后的记忆与恢复机制。

B. 跨任务记忆。 B 组所有对照均保留任务内建图、恢复和验证。清空地图与标记时保留早期任务记录；隐藏这些记录（包括台账与交接记录）时则保留地图。无跨任务记忆保留上述任务局部机制，但移除两种延续形式，使 s-SR 下降 14.7 个百分点，从而分离跨任务复用的贡献。隐藏记录使 s-SR 下降 9.7 个百分点、步数增加 25%；清空空间记忆使 s-SR 下降 12.6 个百分点、步数增加 32%。这些干预迫使后续任务重建路线或重复先前搜索，增加了保留经验本可避免的探索。

C. 恢复交接记录。 为分离重新启动与经验传递的作用，我们保留恢复机制，但移除或替换交接记录。空交接记录达到 72.9% 成功率，略高于无恢复时的 71.1%；使用其他任务的记录则表现更差。重置对话会移除其历史，若缺乏有效交接记录，新尝试也会失去选择不同搜索方案所需的证据。在触发条件和预算不变时，等长普通摘要仍比结构化交接记录低 8.3 个百分点。交接记录会区分已访问地点与有证据排除目标的地点，并保留尚未尝试的选项，从而为下一次尝试选择搜索位置提供依据。

D. 完成检查。 不同于贯穿搜索过程使用的记忆，完成检查仅在智能体提议停止时介入。停止前验证可延长搜索，而停止后认证只会改变后续任务继承的记录。这有助于解释二者对 s-SR 的影响较小（分别为 5.4 和 2.5 个百分点）。认证还将记录准确率提高至 90.7%，而接受所有声明仅为 82.6%；这一收益无法仅由成功率体现（附录 C.5）。

4.4 持续部署

为研究超出基准时程的记忆复用，我们在模拟的多天中，将每栋房屋的十个 GOAT-Bench 游程串联起来。每个游程边界处，机器人会被放置到下一个指定起点，以模拟关机后恢复运行。它必须先借助 SLAM 和已存储的房屋知识重新确定方位，才能复用先前经验。详细案例见附录 D.2。

为检验整合在保留地图与任务记录之外的作用，我们使用 Opus 5，在全部 36 栋房屋上比较 NavHarness 与无整合配置。二者均跨任务保留地图与任务记录，并使用相同的恢复、验证和任务预算。对照组仅禁用运行结束时的整合，因此不会为后续运行构建房屋笔记。将经验整合进长期记忆后，汇总 s-SR 从 72.8% 提升至 80.5%，SPL 从 36.5 提升至 44.3，配对增益分别为 7.7 和 7.8 个百分点（图 3）。除这些定量结果外，我们还检查部署记录，以定性分析智能体如何使用记忆理解目标、引导搜索并重新审视早期结论。

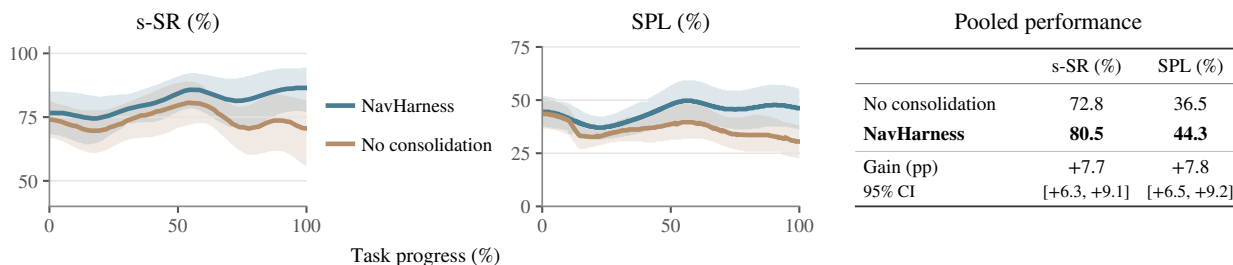


Figure 3 使用 Opus 5 在 36 栋房屋上持续部署，仅改变整合设置。表格报告汇总分数、配对增益和 95% 置信区间。

借助经验理解目标。 在一个重复出现的梳妆台任务中，“楼梯扶手上方”这句话会使未启用整合的智能体前往楼上，但目标实际位于冷冻室。NavHarness 读取一条将相同表述与梳妆台关联的房屋笔记，因此会在正确房间搜索。即使文字本身暗示了不同搜索位置，已存储的关联仍能帮助会话识别预期对象。会话会在移动或查询地图之前阅读该笔记，在规划路线前确定指令所指的楼层和对象。

这些关联不必在首次记录时便得到定论。房屋笔记最初无法确定两种冰箱描述指的是同一台还是两台电器。完成后续任务后，会话重新审视这一问题，检查冰箱两面，并记录答案供后续整合。当之后的任务提供一张昏暗照片时，NavHarness 能识别该电器，而未启用整合的智能体会去地下室寻找第二台冰箱。在记忆中未解决问题的提示下，会话收集了超出当前目标需求的信息；该答案随后帮助未来会话理解新目标。

复用失败尝试并修订记忆。 未成功任务同样能提供有用经验。在镜子搜索中，早期尝试耗尽步数，但记录了敞开门后仍未检查的一面墙。后续会话最初依据不完整的房屋笔记前往错误的镜子。恢复后，它将目标图像与笔记进行比较并搜索日志，从中找到早期建议。即使整合后的笔记未包含该信息，失败尝试仍保留了一个可继续搜索的具体位置。当房屋摘要不足时，详细记录可帮助会话核查已搜索和未检查的位置；随后，会话可依据早期观测选择下一处搜索位置，而非完全依赖整合所保留的内容。

有时只需修订记录的一部分。一条已存储笔记描述了有两株植物的角落，却建议避开该处。当新目标照片要求寻找其中一株植物时，智能体借助描述定位该角落，找到合适视点，并记录一项修正，随后由整合机制纳入笔记。这样既可保留空间知识，又能修订附加其上的建议。类似地，重新启动改变 **SLAM** 坐标系后，地标描述仍可在重建坐标时发挥作用。在这些案例中，推理会话判断记忆的哪些部分适用于当前任务，并提供可供后续会话复用的观测。

记忆可靠性仍是一项局限。判断器接受的错误完成结果可能传播至房屋笔记，而修正一条记录不一定会同步更新其副本，也不能验证其他空间声明。附录 [E](#) 讨论这些尚存风险。

5 结论

NavHarness 使多轮会话能够利用地图、搜索记录和房屋知识，而无须额外训练，从而提升了四种推理模型的导航表现。结构化搜索证据在上一次对话结束后仍可保留，使全新尝试能够从中获益；机器人因此可以重新考虑方案，同时不丢弃其在搜索过程中获得的信息。在扩展部署中，这些经验有助于解释歧义目标、调查未决问题并恢复失败的搜索；与此同时，新观测带来的修正也可供后续会话使用。这些结果支持以连续推理会话为基础构建终身导航系统：后续会话继承有用经验，并在行动过程中持续核验和修订这些经验。

References

- Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents, 2018a. URL <https://arxiv.org/abs/1807.06757>.
- Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018b. URL <https://arxiv.org/abs/1711.07280>.
- Anthropic. Claude Opus 5 system card. <https://www.anthropic.com/transparency>, 2026.
- Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. doi: 10.1109/TRO.2021.3075644.
- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017.
- Peixin Chen, Guoxi Zhang, Jianwei Ma, and Qing Li. Hypothesis graph refinement: Hypothesis-driven exploration with cascade error correction for embodied navigation, 2026a. URL <https://arxiv.org/abs/2604.04108>.
- Yang Chen, Lirong Che, Zhenyu Huang, Wenbo Fu, Chuang Wang, Xu Cao, Daqi Liu, Yuzhe Yang, Jian Su, and Lan-Zhe Guo. HarnessVLN: Unifying training-free embodied navigation through an agent harness, 2026b. URL <https://arxiv.org/abs/2609.15195>.
- Yang Chen, Zhenyu Huang, Wenbo Fu, Danyang Peng, Shi-Yu Tian, Kun-Yang Yu, and Lan-Zhe Guo. STEGNav: Spatio-temporal event graph reasoning for multimodal lifelong object navigation, 2026c. URL <https://arxiv.org/abs/2608.28279>.
- Yanzhe Chen, Zechen Bai, Zhijun Cao, Wenzheng Zeng, Kevin Qinghong Lin, Yiqi Lin, Guoqiang Liang, Kevin Yuchen Ma, Qiming Huang, and Mike Zheng Shou. Show-Harness: Just a VLM agent can play robots, 2026d. URL <https://arxiv.org/abs/2609.10522>.
- Yucheng Deng, Pingrui Lai, Xinhai Li, Chenjia Bai, Xiaoheng Deng, Chengnuo Sun, Xuelong Li, and Hua Yang. SpaceVLN: A zero-shot vision-and-language navigation agent with online spatial cognitive memory and reasoning, 2026. URL <https://arxiv.org/abs/2606.08992>.
- Xin Ding, Liang Mi, Mingzhe Huang, Zixuan Wang, Chao Zhang, Zixu Hao, Fu Chen, Xiangyu Li, Yikai Zheng, Yaoyu Guo, Weijun Wang, Kun Li, Hao Wu, Yunxin Liu, and Ting Cao. Zetta Ç: An efficient closed-loop embodied harness for self-evolving physical intelligence, 2026. URL <https://arxiv.org/abs/2608.16590>.
- Karim Elmaaroufi, Justin Svegliato, Sarunas Kalade, Graham Schelle, Sanjit A. Seshia, and Matei Zaharia. RHO: Your coding agent is secretly a roboticist, 2026. URL <https://arxiv.org/abs/2606.16458>.
- Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- Zuhao Ge, Xiaosong Jia, Chao Wu, Yuchen Zhou, Zuxuan Wu, and Yu-Gang Jiang. EvoMemNav: Efficient self-evolving fine-grained memory for zero-shot embodied navigation, 2026. URL <https://arxiv.org/abs/2606.03509>.
- Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering, 2024. URL <https://arxiv.org/abs/2411.05755>.
- Jianyuan Guo, Zhiwei Hao, Chengcheng Wang, Cheng Fan, Tingzhang Luo, Hongguang Li, Ying Gao, Hefei Mei, Jiankun Peng, Rongjian Xu, Mingjing Dong, Han Wu, Mengyu Zheng, Kai Han, Shiqi Wang, Chang Xu, and Yunhe Wang. From question answering to task completion: A survey on agent system and harness design, 2026. URL <https://arxiv.org/abs/2606.20683>.
- Zebin Han, Xudong Wang, Baichen Liu, Qi Lyu, Zhenduo Shang, Jiahua Dong, Lianqing Liu, and Zhi Han. SeqWalker: Sequential-horizon vision-and-language navigation with hierarchical planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL <https://arxiv.org/abs/2601.04699>.
- Xubin Hao, Hongjin Meng, Xin Yin, Jiawei Zhu, and Chenpeng Cao. Self-GC: Self-governing context for long-horizon LLM agents, 2026. URL <https://arxiv.org/abs/2607.00692>.

- Jinbang Huang, Yuanzhao Hu, Zhiyuan Li, Ran Qi, Yixin Xiao, Zhanguang Zhang, Mark Coates, Tongtong Cao, and Yingxue Zhang. RoboHarness: Memory-driven orchestration of heterogeneous robot policies for long-horizon planning, 2026. URL <https://arxiv.org/abs/2607.18060>.
- Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping, 2019. URL <https://arxiv.org/abs/1907.05446>.
- MoniJesu James, Marcelino Julio Fernando, Miguel Altamirano Cabrera, and Dzmitry Tsetserukou. AnyGoal: Vision-language guided multi-agent exploration for training-free lifelong navigation, 2026. URL <https://arxiv.org/abs/2606.13878>.
- Zihe Ji, Huangxuan Lin, and Yue Gao. DyNaVLM: Zero-shot vision-language navigation system with dynamic viewpoints and self-refining graph memory, 2025. URL <https://arxiv.org/abs/2506.15096>.
- Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A. Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing context compression for long-horizon LLM agents. In *International Conference on Machine Learning (ICML)*, 2026. URL <https://arxiv.org/abs/2510.00615>.
- Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A benchmark for multi-modal life-long navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Khanna_GOAT-Bench_A_Benchmark_for_Multi-Modal_Lifelong_Navigation_CVPR_2024_paper.html.
- Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020.
- Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. Iterative vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14921–14930, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Krantz_Iterative_Vision-and-Language_Navigation_CVPR_2023_paper.html.
- Taekbeom Lee, Youngseok Jang, Jeonghwa Heo, Jeongjun Choi, and H. Jin Kim. ObsGraph: Hierarchical observation representation for embodied reasoning and exploration, 2026. URL <https://arxiv.org/abs/2606.24068>.
- Zixing Lei, Gengze Zhou, Xiong-Hui Chen, Jiazhao Zhang, Yiyang Huang, Hang Yin, Haoqi Yuan, Qi Wu, Weixin Li, and Siheng Chen. Scaffolding foundation models into physical-world agents pushes the frontier of long-horizon navigation, 2026. URL <https://arxiv.org/abs/2608.30396>.
- Ji Li, Bo Wang, Jing Xia, Mingyi Li, and Shiyang Hu. HIMM: Human-inspired long-term memory modeling for embodied exploration and question answering, 2026a. URL <https://arxiv.org/abs/2602.15513>.
- Xiaochuan Li, Ryan Ming, Meng Chu, Shuai Shao, Rong Jin, and Chenyan Xiong. ACM: Agentic context management for long horizon tasks, 2026b. URL <https://arxiv.org/abs/2607.23809>.
- Xueying Li, Feng Lyu, Hao Wu, Mingliu Liu, Jia-Nan Liu, and Guozi Liu. Stop wandering: Efficient vision-language navigation via metacognitive reasoning, 2026c. URL <https://arxiv.org/abs/2604.02318>.
- Yijian Li, Changze Li, Hantian Shi, Jiaying Luo, Jiyan Cai, Ming Yang, and Tong Qin. AgenticNav: Zero-shot vision-and-language navigation as a tool-calling harness, 2026d. URL <https://arxiv.org/abs/2606.10577>.
- Yiren Lu, Yi Du, Disheng Liu, Yunlai Zhou, Chen Wang, and Yu Yin. GSMem: 3D gaussian splatting as persistent spatial memory for zero-shot embodied exploration and reasoning, 2026. URL <https://arxiv.org/abs/2603.19137>.
- Qianyu Meng, Yanan Wang, Liyi Chen, Yihang Li, Wei Wu, Wenyuan Jiang, Qimeng Wang, Chengqiang Lu, Yan Gao, Yi Wu, and Yao Hu. Agent harness for large language model agents: A survey. Preprints.org, 2026. URL <https://doi.org/10.20944/preprints202604.0428.v3>. Version 3.
- Haochen Niu, Lantao Zhang, Xingwu Ji, Rendong Ying, Peilin Liu, and Fei Wen. SSMG-Nav: Enhancing lifelong object navigation with semantic skeleton memory graph, 2026. URL <https://arxiv.org/abs/2603.01813>.
- OpenAI. GPT-4o system card, 2024. URL <https://openai.com/index/gpt-4o-system-card/>.
- OpenAI. GPT-6 Astra: A new generation of intelligence. <https://openai.com/index/gpt-6-astra/>, 2026.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL <https://arxiv.org/abs/2310.08560>.
- Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL <https://arxiv.org/abs/2304.03442>.

- Qwen Team. Qwen3.8-27B model card. <https://huggingface.co/Qwen/Qwen3.8-27B>, 2026.
- Prithvi Rajasekaran. Harness design for long-running application development. Anthropic Engineering, 2026. URL <https://www.anthropic.com/engineering/harness-design-long-running-apps>.
- Prithvi Rajasekaran, Ethan Dixon, Carly Ryan, and Jeremy Hadfield. Effective context engineering for AI agents. Anthropic Engineering, 2025. URL <https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents>.
- Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Under-sander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. In *NeurIPS Datasets and Benchmarks Track*, 2021.
- Botao Ren, Junjun Hu, Xinda Xue, Minghua Luo, Jintao Chen, Haochen Bai, Liangliang You, and Mu Xu. AstraNav-Memory: Contexts compression for long memory, 2025. URL <https://arxiv.org/abs/2512.21627>.
- Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- Andrew Semenov and Svyatoslav Dorofeev. Beyond compaction: Structured context eviction for long-horizon agents, 2026. URL <https://arxiv.org/abs/2606.11213>.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL <https://arxiv.org/abs/2303.11366>.
- Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. URL <https://arxiv.org/abs/2309.02427>.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL <https://openreview.net/forum?id=pAMNKGwja6>.
- Peidong Wang, Zhiming Ma, Ying Chang, Xufang Luo, Yiqun Zhang, Zihan Wang, Xiaocui Yang, Shi Feng, Yuqing Yang, and Dongsheng Li. Self-evolving embodied agents via skill-harness evolution, 2026a. URL <https://arxiv.org/abs/2608.11350>.
- Qi Wang, Tianyi Wang, Chengyang Li, Shikun Ban, Yurun Chen, Yizhong Ge, Jason Qin, Chengtai Li, and Wentao Zhu. Towards the harness of embodied agents, 2026b. URL <https://arxiv.org/abs/2608.11246>.
- Sen Wang, Bangwei Liu, Zhenkun Gao, Lizhuang Ma, Xuhong Wang, Yuan Xie, and Xin Tan. Explore with long-term memory: A benchmark and multimodal LLM-based reinforcement learning framework for embodied exploration, 2026c. URL <https://arxiv.org/abs/2601.10744>. CVPR 2026.
- Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2407.16741>.
- Xudong Wang, Jiahua Dong, Baichen Liu, Qi Lyu, Lianqing Liu, and Zhi Han. Lifelong embodied navigation learning. In *International Conference on Learning Representations (ICLR)*, 2026d. URL <https://openreview.net/forum?id=PaYo96rjjj>.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL <https://arxiv.org/abs/2502.12110>.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024. URL <https://arxiv.org/abs/2405.15793>.
- Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3D-Mem: 3D scene memory for embodied exploration and reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- Hang Yin, Yinan Liang, Jiazhaoh Zhang, Jiahang Liu, Minghan Li, Zhizheng Zhang, and He Wang. AllDayNav: Lifelong navigation via real-world reinforcement learning, 2026. URL <https://arxiv.org/abs/2606.10927>.
- Justin Young. Effective harnesses for long-running agents. Anthropic Engineering, 2025. URL <https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents>.

- Gengyuan Zhang, Mingcong Ding, Jingpei Wu, Ruotong Liao, and Volker Tresp. ReEXplore: Improving MLLMs for embodied exploration with contextualized retrospective experience replay, 2025a. URL <https://arxiv.org/abs/2511.19033>.
- Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models, 2025b. URL <https://arxiv.org/abs/2510.04618>.
- Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Chunyang Zhu, Jiaying Qiu, Yuchen Yan, Jiayuan Liu, Wenhao Tang, Zhengru Fang, Yi Nie, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents, 2026. URL <https://arxiv.org/abs/2607.08448>.
- Ganlong Zhao, Guanbin Li, Weikai Chen, and Yizhou Yu. OVER-NAV: Elevating iterative vision-and-language navigation with open-vocabulary detection and structured representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL <https://arxiv.org/abs/2403.17334>.
- Shibo Zhao, Guofei Chen, Honghao Zhu, Zhiheng Li, Changwei Yao, Nader Zantout, Seungchan Kim, Wenshan Wang, Ji Zhang, and Sebastian Scherer. SuperMap: A spatio-temporal SLAM system for visual-language navigation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2026. URL <https://superodometry.com/supermap>.
- Jian Zhou, Xunyi Zhao, Gengze Zhou, Zerui Li, Sihao Lin, Jiajun Liu, and Qi Wu. Embodied agents take control: Minimal-interface zero-shot agents rival industrial-scale policies in vision-and-language navigation, 2026. URL <https://arxiv.org/abs/2607.26148>.
- Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, Siyuan Huang, and Qing Li. Move to understand a 3D scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.
- Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. TANGO: Training-free embodied AI agents for open-world tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

附录

本附录补充实现细节，并对实验结果作进一步考察。附录 A 给出完整算法、六组件执行框架规范和状态转移规则；附录 B 介绍评测设置、统计流程与补充对照实验；附录 C 报告组件实验结果和诊断分析；附录 D 考察扩展部署与记忆使用案例；附录 E 讨论局限性和未来方向。

A 执行框架规范

算法 1 NavHarness 在 K 个任务序列上的运行流程

Require: 目标 $g_{1:K}$ 、导航与记忆工具、恢复控制器、判定器、任务限制及步数上限 H

```
1: 挂载空间记忆、任务记录及已有房屋笔记
2: for 每个目标  $g_k$  do
3:   写入目标，并准备引用可用记忆文件的指令
4:   if 机器人自上一任务后已被重定位 then
5:     附上周围视图以及识别当前位置的指令
6:   end if
7:   打开全新的导航会话，并初始化任务计数器与闭合声明
8:   while 任务仍处于打开状态且执行预算尚有剩余 do
9:     推进会话至多  $H$  个模型轮次，包括工具调用
10:    从工具桥读取执行状态和待处理请求
11:    if 环境已终止任务 then
12:      跳出
13:    else if 会话已请求任务闭合 then
14:      if 任务交接记录缺失且尚未请求过 then
15:        要求会话写入交接记录；继续
16:      end if
17:      if 这是首个进入验证的闭合请求 then
18:        根据目标、视图、地图和执行记录获取判定器结论
19:        if 判定明确且与导航器的闭合声明矛盾 then
20:          将反证返回同一会话；继续
21:        end if
22:      end if
23:      发出 STOP 并闭合任务
24:    else
25:      由恢复控制器评估请求或已启用的时长回退条件
26:      if 恢复获准 then
27:        让退出会话在最后一个仅文件步骤中写入恢复笔记
28:        准备所选范围的重新评估及位置简报（如有）
29:        保留目标、物理状态、地图、任务记录和剩余预算
30:        用全新会话替换当前对话，并指示其读取恢复笔记
31:      else
32:        恢复同一会话，除非持续无活动要求终止
33:      end if
34:    end if
35:  end while
36:  确保任务已经结束，并记录由预算耗尽或无活动导致的终止
37:  认证最终结果；若动作步数未变，则复用停止前判定
38:  将声明、判定、可用任务交接记录及命名地点汇入任务记录
39:  写入判定和台账条目，更新地点记忆并保存运行检查点
40:  在打开下一任务会话前释放执行对话
41: end for
42: 关闭执行会话
43: if 已启用跨运行整合 then
44:   由独立的仅文件会话将日志整合到四个房屋笔记文件中
45: end if
```

NavHarness 包含六个组件：循环、工具注册表、上下文策略、记忆、在边界触发的钩子，以及对完成声明的验证（图 4）。这一划分沿用 Meng et al. (2026)；Guo et al. (2026) 也区分观测、动作与治理接口。本文以这六个组件组织实现。NavHarness 规定导航边界保留的状态以及尝试之间的交接记录，详见下文。



Figure 4 NavHarness 的六个组件及其职责。

A.1 循环

编码智能体会话是 NavHarness 的推理核心。它通过多轮多模态对话解释观测、查阅记忆，并调用导航与文件工具。外层编排器管理任务序列，并决定会话何时继续、闭合，或将其经验交接给新的尝试。算法 1 描述了这一生命周期。工具执行仍在编码会话内部进行，因此编排器无须另行实现模型推理与工具执行循环。

对于房屋 h 中的一次部署，将任务 k 闭合时的状态记为

$$X_k = (C_k, M_k, L_k, D_h), \quad M_k = (\{G^{(f)}\}_f, \mathcal{P}_k, \Gamma_k), \quad L_k = (r_1, \dots, r_k), \quad (1)$$

其中， C_k 是活动会话上下文， $G^{(f)}$ 是楼层 f 的占用栅格， $\mathcal{P}_k$ 是命名地点及其视图， Γ_k 是地点图， L_k 是任务记录， D_h 是持久房屋笔记。在一次尝试内，常规交互按下式扩展对话：

$$C_{k,a,t+1} = C_{k,a,t} \parallel (y_t, \mathcal{A}_t, o_{t+1}), \quad (y_t, \mathcal{A}_t) = \pi_\theta(C_{k,a,t}, \mathcal{T}), \quad (2)$$

其中 y_t 为回复， $\mathcal{A}_t$ 为工具调用， o_{t+1} 为返回观测； a 索引尝试， t 索引交互。NavHarness 不重写活动对话，且为每项任务打开全新会话，因此对话记录不会跨越任务边界。只有单会话对比配置会在任务之间保留同一对话。

任务 k 闭合前，只有 L_{k-1} 可作为先前任务历史，因为 r_k 要到闭合时才附加。因此，式 7 中的会话打开过程引用 L_{k-1} 。

编排器以步段推进会话。每个步段允许至多 H 个模型轮次及其工具调用，并返回所得执行状态。多个步段可属于同一对话。步段返回后，环境终止的优先级最高，其次是待处理的闭合请求，最后是恢复；若均未发生，编排器以继续指令恢复会话。闭合与恢复工具会记录请求并要求智能体让出控制，从而在步段之间处理这些转换。任务在不同尝试之间保留轮次和物理步数限制 (B, b) 。即使没有闭合声明，预算耗尽或持续无活动也会终止任务。实现默认每个步段 $H = 25$ 个模型轮次，最早可在尝试的第 12 个轮次请求恢复。连续三个步段未调用工具会终止无活动任务；若三个步段均没有模型轮次，则抛出提供方不可用错误，而不记录为导航失败。任务预算与已启用的回退设置见附录 B。

会话适配器向编排器提供打开、推进和关闭对话的接口，其工具桥将编码循环连接到环境与记忆工作区。只要后端支持会话和工具接口，这种分离就允许不同推理模型复用相同的记忆与边界规则。执行、判定与整合使用彼此独立的对话和工具权限（附录 A.3 与 A.6）。

A.2 工具注册表

模型仅通过表 4 中的调用执行动作。导航器提供移动、建图与路线查询，具身执行框架则增加文件访问以及闭合和恢复请求。空间调用所依赖的定位工具将在表后说明。

感知与运动。 `observe` 返回前向相机图像。`step` 接收离散动作列表，包括前进 0.25 m、左转 15° 或右转 15°，并按顺序执行。每个动作消耗任务物理预算中的一步，会话还受模型轮次上限约束。模型会被告知，原地转向是一种低成本环视方式，并应交替进行观察与移动。

一次 `step` 调用最多接收十二个动作 $\alpha = (a_1, \dots, a_n)$ ，动作集合为 $\mathcal{A} = \{\text{forward 0.25 m, left 15}^\circ, \text{right 15}^\circ\}$ 。每执行一个动作消耗一步，并拆分为至多 $5n$ 个 0.05 m 或 3° 的子运动；每个子运动向定位工具馈送一帧。环境终止任务时，执行提前结束。面向智能体的其他调用均不消耗步数。

动作离散化与碰撞。 仿真器分别执行每个子运动；若检测到碰撞，宏动作会提前结束。仿真器的具身配置启用了滑动。这一细化为 SLAM 提供所需的中间观测。组件与会话比较均使用这一运动接口，并保留基准目标与评分方式。

Table 4 工具注册表。只有 step 消耗任务的物理预算，免费调用不消耗步数。

调用	参数	返回值或作用
observe	无	前向 RGB 视图及剩余步数预算
step	动作列表	执行动作并报告运动、碰撞和预算状态
get_goal	无	当前目标，以文本或目标图像形式返回
get_map	楼层，缩放级别	带标记的俯视占用地图
mark	地点名称	在当前位置放置或移动命名标记
preview_path	地图点	已知自由空间上的路线长度；不发生移动
list_files	目录	列出可用记忆文件
read_file	文件路径	读取任务记录或房屋笔记
write_file	文件路径，文本	写入分配给本会话的文件
request_recovery	范围，原因，置信度	提议恢复；由具身执行框架决定
close_task	状态，证据	捕获停止视图并请求经检查的闭合

目标。get_goal 重新读取当前目标。若目标以照片形式给出，该调用会返回照片，因此即使其他图像已进入上下文，仍可再次查阅图像目标。

空间记忆。get_map 返回导航器随本体移动构建的占用地图及其命名标记。地图跨任务持久保留，使后续搜索能够复用已探索空间和先前地标。mark 放置以地点而非搜索目标物体命名的标记；具身执行框架为其附上在该点拍摄的视图，按朝向分桶且最多保留四幅，并记录导航器的位姿估计。用已有名称再次标记会移动该标记。preview_path 在地图的已知自由空间上绘制到某点的路线并报告长度，而不移动本体。三种调用均免费。

时刻 t 以名称 a 调用 mark 时，会替换所有同名地点：

$$\mathcal{P} \leftarrow (\mathcal{P} \setminus \{z_i : a_i = a\}) \cup \{(a, m_t, B_t)\}, \quad |B_t| \leq 4, \quad (3)$$

其中， m_t 是当前估计位姿处的标记， B_t 是在该处拍摄并按朝向分桶的视图。get_map 根据请求的楼层与缩放级别返回 $\text{Render}(G^{(f)}, \mathcal{P}^{(f)})$ ，preview_path 则在 $G^{(f)}$ 上计算式 6。

定位。observe、get_map、mark 和 preview_path 背后共用一个估计器，本文将其视为与其他工具相同的工具。报告的运行使用 RGB-D 模式的 ORB-SLAM3 (Campos et al., 2021)，它作为独立进程运行，由导航器向其馈送数据并查询。具身执行框架不依赖这一选择。该工具接收每个子运动的彩色帧与深度帧，返回位姿估计及状态：

$$(\hat{\mathbf{T}}_t, s_t) = \text{SLAM}(I_t, D_t \mid \hat{\mathbf{T}}_{<t}), \quad \hat{\mathbf{T}}_t \in SE(3), \quad s_t \in \{\text{ok}, \text{lost}\}, \quad (4)$$

世界坐标系锚定于在房屋内看到的第一帧。从 $\hat{\mathbf{T}}_t$ 读出的平面位姿 $(\hat{x}_t, \hat{z}_t, \hat{\psi}_t)$ 是导航器拥有的唯一位置信息；标记存储该位姿，地图在该坐标系中绘制，路线也从该位姿开始规划。这些过程从不读取仿真器位姿。

每个粗粒度动作——前进 0.25 m 或转向 15° ——均拆分为五个 0.05 m 或 3° 的子运动，每个子运动以合成的 10 Hz 频率产生一个跟踪帧。拆分是为了跟踪器而非本体：每次 15° 转向仅提供一帧会导致特征匹配过少，使大量帧丢失跟踪；每帧转向 3° 则能在导航期间维持帧间匹配。若运行中丢失跟踪，即 s_t 为 lost，则不返回位姿且不写入地图。跟踪状态通过运动响应、地图响应以及工具桥的执行状态暴露。跟踪器在下次成功初始化时打开新地图，导航器将其重新锚定到最后一可信位姿： $\hat{\mathbf{T}} \leftarrow \mathbf{A} \hat{\mathbf{T}}$ ，其中 $\mathbf{A} = \hat{\mathbf{T}}_{\text{last}} \hat{\mathbf{T}}_{\text{first}}^{-1}$ 。受控重定位后，本体原地转动一周，同时向跟踪器馈送数据，以尝试相对于保留的跟踪器地图进行重定位。跟踪器重启后加载已保存的占用地图与标记，本身并不能恢复坐标对齐。

建图与路线。占用地图根据深度和估计位姿构建，而非使用跟踪器自身的点云，因此对于任何能返回位姿的估计器，计算方式均相同。将满足 $D_t(u) \in [0.3, 4.5]$ m 的深度像素 u 反投影到世界中，并读取其相对于相机下方地面的高度：

$$\mathbf{p} = \hat{\mathbf{T}}_t \pi^{-1}(u, D_t(u)), \quad h(\mathbf{p}) = h_c - (p_y - y_{\text{ref}}), \quad (5)$$

建图器采用向下为正的 y 轴。 y_{ref} 是相机在该楼层上的参考世界 y 坐标， h_c 是相机高出楼面的高度。因此，楼面位于 $y_{\text{ref}} + h_c$ ，此处公式给出的高度为零。当 $0.15 < h < 1.4 \text{ m}$ 时，点被视为障碍物；当 $-0.5 < h \leq 0.15 \text{ m}$ 时，则视为地面。点落入边长 $\delta = 0.10 \text{ m}$ 的栅格单元， $c(\mathbf{p}) = \lfloor (p_x, p_z) / \delta \rfloor$ 。单元有未知、自由与障碍物三种状态：障碍物端点标记其所在单元，从本体到该端点的射线刻画自由单元，且障碍物不会被自由状态覆盖。每个楼层分别保存栅格；当估计高度稳定在距所有已注册楼层超过 1 m 的位置时，打开新图层。`get_map` 渲染带标记的当前图层。栅格依据在线位姿写入，因此跟踪器之后的回环闭合校正不会传播到已写入的单元，这是附录 E 所述局限之一。

`preview_path` 在同一栅格上回答路线查询。若目标单元 c^* 落在未知或障碍物区域，则将其吸附到最近的自由单元；随后在自由单元上以八邻接移动运行 A^* ，并报告

$$L(c^*) = \min_{\pi: c_0 \rightarrow c^*} \sum_i \delta \|c_{i+1} - c_i\|_2, \quad (6)$$

若不存在这类路径，则报告不可达。此过程不发生移动。标记由名称、调用时刻的平面位姿 $m_k = (\text{name}_k, (\hat{x}, \hat{z}, \hat{\psi})_{t_k})$ 及在该处存储的视图组成。

替换估计器。定位估计器提供位姿与跟踪状态。独立的建图适配器将其与深度结合，获得占用地图使用的世界点。若其他视觉或视觉惯性估计器能提供兼容位姿，也可使用同一适配器。平面激光雷达需要不同的距离投影与障碍物更新方式，但可保留导航工具和编排器接口。因此，更换传感器可能要求修改工具侧实现，但无须改变会话与记忆生命周期。本文报告的运行使用 ORB-SLAM3。

闭合任务。智能体在调用 `close_task(c_k, \epsilon_k)` 前写入交接记录，其中 $c_k \in \{\text{complete}, \text{incomplete}\}$ ， ϵ_k 是对停止证据的简短说明。该工具捕获周围视图、记录请求并要求会话让出控制；它本身不会发出 **STOP**。在下一个步段边界，若交接文件缺失，编排器会提示一次，随后应用附录 A.6 的停止前验证。获准请求会转化为物理 **STOP**。任务也可能因执行限制而结束，此时台账记录终止情况，而不会虚构完成声明。

恢复。`request_recovery` 允许模型指出其部分依据信息已不再可信，并通过命名满足条件的最小范围说明受影响程度（附录 A.5）。该调用不消耗步数，也不终止任务。具身执行框架决定后续动作：可以拒绝，也可以采用比请求更大的范围。

提议是三元组 $(\hat{u}, \text{reason}, c)$ ：范围 $\hat{u} \in \{\text{attempt}, \text{place}, \text{room}, \text{floor}, \text{house}\}$ 、文本原因以及置信度 $c \in [0, 1]$ 。模型从不提议继续；这是具身执行框架依据式 14 作出的答复。

文件访问。会话通过 `list_files`、`read_file` 和 `write_file` 访问文本记忆。路径在运行工作区或其挂载的房屋目录中解析。执行会话可读取先前记录，但只能写入分配给它的交接记录与恢复笔记。编排器拥有台账和判定文件，整合会话则拥有房屋笔记。会话写入的叙述会清除形似坐标的表达，避免地图之后重新锚定时在文本中遗留过时的度量坐标。一次读取最多返回 20,000 个字符，一次写入最多存储 6,000 个字符，工具会报告截断。这些是文件访问限制，与模型上下文窗口相互独立。当前读取工具没有偏移参数：过长文件只返回其前缀。详细任务文件可通过目录列表定位并逐个读取，但单次读取台账不能保证访问任意长的历史记录。

A.3 上下文管理

每次尝试一个会话。任务边界策略使用相同的常设系统简报 σ 打开全新编码会话。第一条用户指令给出目标、指定要读取的文件，并标明允许写入的交接文件。如这些文件存在，必读内容包括上一任务的交接记录、运行台账和已有房屋笔记。会话通过自身的文件读取将其内容纳入对话，而不是将内容粘贴到系统消息中。

将这些文件引用记为 ρ_k ，会话打开过程为

$$C_{k,1,0} = \text{Open}(\sigma, I(g_k, \rho_k)), \quad \rho_k = \text{refs}(h_{k-1}, L_{k-1}, D_h), \quad (7)$$

其中不存在的文件会被省略。图像目标仍可通过 `get_goal` 访问，地图则可通过空间工具访问。会话被要求在移动前读取列出的记录，并可随任务推进查阅其他文件。之后，会话跨步段持续运行，具身执行框架不会编辑对话记录。

任务与尝试边界处的重置。在任务边界，编排器记录闭合状态，并使执行者写入的交接记录可供下一会话使用。恢复获准时，当前会话先写入恢复笔记，再由新会话依据该文件恢复同一任务。两种转换都不会将旧对话带入新会话。地图状态、任务记录和房屋笔记保留在会话外部，可再次读取。恢复会保留物理轨迹与剩余任务预算。

上下文管理的职责划分。NavHarness 控制对话何时结束，编码后端控制如何维护进行中的对话。系统不引入图像逐出窗口或额外的对话记录摘要策略。将这些职责分开，便可在不改变记忆接口的情况下替换推理核心。提供方支持时，未改变的前缀也允许复用提示缓存。导航轮次预算适用于执行会话，判定与运行结束时的整合使用独立模型调用。

会话策略与观测到的上下文使用。独立会话基线为每项任务打开全新对话，且仅依赖该上下文，不使用外部工作记忆或长期记忆。单会话策略在每项任务后继续使用同一对话，不断累积观测和工具响应，直至后端上下文窗口填满；随后后端压缩对话以继续运行。任务边界不会触发压缩。NavHarness 则在任务边界打开全新对话，同时保留外部记忆，平均峰值提示约为 20k 个词元。表 3 比较这些完整策略；其中的记忆干预保持其他具身执行框架机制不变。

连续任务中的成功率。图 5 比较 NavHarness 与单会话策略在序列位置 1 至 10 上拟合的成功率趋势。这些曲线概括任务在该范围内推进时的性能。配对系统比较及其不确定性见表 3。由于回合长度不同，靠后位置符合条件的场景更少。这些汇总区间不是曲线的置信带。表 5 另行报告 NavHarness 在 GOAT-Bench Val-Unseen 任务位置审计中实测的位置级成功率。这些是单次运行的诊断性观测，而非三随机种子组件研究的平均值。

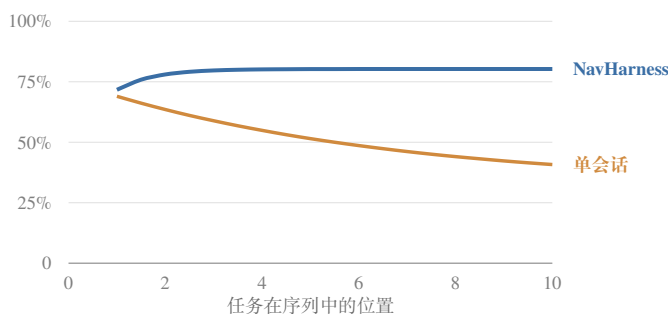


Figure 5 连续任务中的成功率。NavHarness 与单会话在位置 1 至 10 上的拟合趋势。

Table 5 NavHarness Val-Unseen 诊断运行中观测到的任务位置成功率。

位置	1	2	3	4	5	6	7	8	9	10
s-SR (%)	72.2	72.2	83.3	83.3	77.8	82.1	73.1	85.7	81.2	85.7

上下文占用与输入成本扩展。图 6 展示一次更长期部署的前 20 项任务。在上下文示意面板中，单会话策略在接近任务 20 时达到约 1M 词元的窗口，此时后端压缩对话并继续运行。该边界不会结束部署。NavHarness 则打开任务级会话，其活动上下文约为 20k 个词元。两个面板均以灰色表示单会话。

成本面板区分保留会话与无状态上下文重建；后者在每次交互时重建提示。橙色曲线表示后者缓存未命中的情况，而非单会话基线或独立会话基线。保留稳定前缀可复用缓存，重建提示则可能使该前缀失效。因此，实际输入成本同时取决于未缓存与已缓存输入量： $C_K = \sum_{k=1}^K (p_u U_k + p_c V_k)$ ，其中 U_k 与 V_k 分别是任务 k 对应的词元数。

成本比较假设每百万输入词元价格为 \$5。任务 k 使用 $2.4 \times 10^6 k$ 个输入词元时，无缓存命中的无状态重建在 20 项任务上的成本约为 \$2,520。NavHarness 每项任务使用 0.975M 个输入词元，按相同价格计算约为 \$98。对保留会话，本文采用线性参考 $C_K = 528K/20$ 美元，在任务 20 达到 \$528。这些计算仅涵盖导航输入词元，不包括输出费用与辅助调用，并止于任务 20。基准运行时间见附录 B.3。

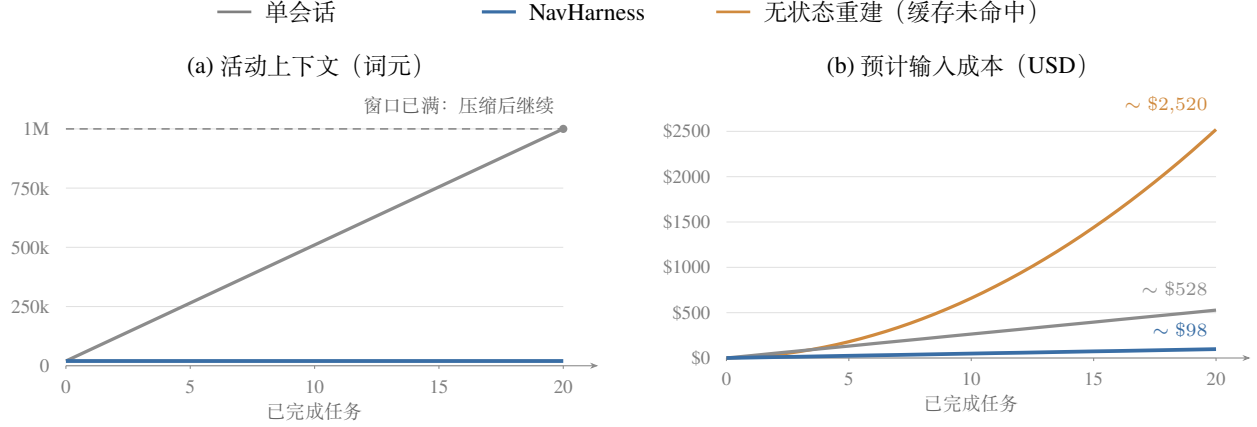


Figure 6 更长期部署前 20 项任务的上下文与输入成本扩展。橙色表示无缓存命中的无状态上下文重建。

A.4 记忆

记忆按生命周期与访问方式组织，而非划分为互不相交的存储位置。编码会话维护当前对话。工作记忆包含传感器构建的地图、命名地点以及连续任务间共享的任务记录。地图与地点记录若被保留供后续运行使用，也会与文本笔记和经验存档一同构成长期房屋记忆。因此，同一地图无须复制进文本摘要即可同时承担两种角色。

台账与任务交接记录。本文用交接记录表示跨会话边界传递的经验。任务交接记录在闭合时记录任务，恢复笔记则将搜索证据传递给同一目标的新尝试（附录 A.5）。二者是独立文件，如表 6 所示。请求闭合前，执行会话写入 h_k ，简要记录目标、到访的房间与地标、结果证据以及未成功的搜索；提示要求其少于 120 个词。编排器在任务边界读取该文件，并将其与声明、检查判定和命名地点组合：

$$r_k = (g_k, c_k, v_k, h_k, p_k), \quad \text{status}(r_k) = \begin{cases} \text{CERTIFIED} & v_k = \text{complete}, \\ \text{NOT DONE} & v_k = \text{incomplete}, \\ \text{UNRESOLVED} & v_k = \text{unknown}, \\ \text{CLAIMED} & \text{no certification ran.} \end{cases} \quad (8)$$

完整运行记录还保留声明与判定证据、已搜索区域、步数及终止原因。环境分数单独存储用于评测，不会呈现在面向智能体的记忆中。认证完成针对目标谓词，因此执行者对路线的描述与空间解释仍属于其自身陈述。**CLAIMED** 表示实现中的未检查状态，而非成功声明。任务在没有模型闭合声明的情况下结束时，记录使用 **stop** 并保留终止原因。即使关闭认证，它仍与显式 **complete** 声明有别。

台账文件是紧凑索引，每项已闭合任务占一行，包含其状态、缩略目标以及最多三个标记名称。详细交接记录与判定保存在独立文件中。后续会话读取台账以确定搜索方向，并可打开相关任务目录获取细节。这样既可单独访问每项任务的记录，也无须将所有记录复制到新对话。完整任务记录 r_k 包含交接记录与判定，台账文件则提供其简短索引条目。台账索引本身会随任务数量增长。

房屋目录从持久存储挂载，运行目录则保存当前序列。重复恢复使用不同的笔记文件名。附录 A.2 的权限检查可防止执行会话覆盖台账、判定器结论或已整合的房屋笔记。

地点。一个地点由标记、在该处拍摄并按朝向分桶的至多四幅视图，以及导航器对拍摄位置的估计组成。视图是观测，位置是估计，名称是声明；三者彼此分开。空间恢复期间，可将当前视图与这些存储观测比较，以识别熟悉地点。形式上，命名地点为

$$z_i = (a_i, m_i, \{(I_j, \hat{\xi}_j)\}_{j \in B_i}), \quad |B_i| \leq 4. \quad (9)$$

其中， a_i 是标签， m_i 是标记， B_i 是存储的视图及其 SLAM 位姿估计 $\hat{\xi}_j$ 。

Table 6 工作区中的文本记忆。每个文件都有指定写入者。

路径	写入者	内容
run/ledger.md	编排器	任务状态索引
run/tasks/k/goal.md	编排器	目标与模态
run/tasks/k/handover.md	执行会话	任务经验
run/tasks/k/lambda.md	执行会话	恢复交接记录
run/tasks/k/verdict.json	编排器	判定器评估
house/	整合会话	房屋笔记：索引、概览、房间笔记、技能

这些地点构成图：

$$\Gamma = (V_{\text{place}}, E_{\text{conn}}),$$

$$E_{\text{conn}} = \{(z_i, z_j) : z_i, z_j \text{ marked consecutively, } f(z_i) \neq f(z_j)\}, \quad (10)$$

其中，每条地点记录带有房间标签和楼层关联 $f(z)$ 。房间标签默认采用标记名称。这些是地点属性，而非独立且全连接的房间—楼层图。房屋笔记中描述的房间连接仍属于文本知识。该图根据任务边界的标记记录和报告的地图图层更新。在不同图层上连续注册的地点之间的边，是供规划使用的候选连接，而非直接观测到的楼梯。图地址由标签和关系构成，度量几何则保留在空间地图中。

房屋存档。长期记忆是为了一栋房屋维护的持久知识，而非单个文件。其文本部分即房屋笔记，由四个文件组成，分别包含索引、房屋概览、房间笔记和导航技能；公式中将其统记为 D_h 。四个文件均仅由整合会话写入。已保存地图、地点记录和经验存档保留各自的表示形式。

持久记录无须等待整合。任务闭合时，编排器更新已启用的地点与经验存储，并在启用检查点时保存空间状态。整合则总结整个任务序列中发生的事件。它读取已有房屋笔记、台账与交接记录，保留有用路线、失败搜索、修正与未解决问题，使后续会话无须重读每条记录。当前实现中，这一独立会话在运行闭合时启动，房屋笔记更新为

$$D_h^+ = \mathcal{C}_\theta(D_h, L_K), \quad (11)$$

其中， L_K 包含可通过文件访问的任务记录。提示要求模型区分认证结果与机器人自身解释，且仅在日志提供支持时才添加导航技能。工作区强制执行文件写入权限，而这些内容要求则以指令形式提供给模型。

整合会话拥有文件工具，但没有导航工具，其写入权限仅覆盖房屋笔记。它可以总结房间与地标描述，但不会重建或修改地图几何。房屋笔记会出现在后续运行的必读列表中，详细记录则仍可单独访问。摘要间隔是一项调度选择：同一操作可在一项任务、一组任务或更长期部署后执行。本文以运行闭合作为调用点，并不要求一次性写入全部长期记忆。存储或总结某项声明并不等同于认证它。

A.5 钩子

钩子是编排器对边界事件的响应。会话可通过工具请求闭合或恢复，但由编排器在步段返回后决定并执行转换。四类事件将推理会话连接到持久记忆。

任务边界。闭合后，编排器获取最终检查判定，读取执行者已写入的交接记录，并附加任务记录。它写入判定文件与台账索引，更新已启用的命名地点和经验存储，并在启用检查点时保存空间状态。下一项任务的指令会引用这些新记录。采用任务边界会话策略时，转换为

$$(C, M, L, D_h) \mapsto (\emptyset, M^+, L \parallel r_k, D_h), \quad (12)$$

其中， M^+ 包含更新后的地点图。即使其他持久记录得到更新，房屋笔记 D_h 在该边界也保持不变。因执行限制而结束的任务遵循相同记录路径，即使它没有交接记录或完成声明。

尝试边界。恢复由模型请求或已启用的时长回退触发。控制器检查是否需要新尝试，再选择重新审视当前解释的范围。所有范围都保留地图、标记、台账与存档。范围越大，对话重置时重新评估的位置或路线越多（表 7），因此重启不会抹去机器人已收集的观测。

控制器旨在选择与观测到的失败相符的最小范围：

$$u_t^* = \min_{u \in \mathcal{U}} \{u : I(u) \text{ covers the beliefs implicated by the observed failure}\}, \quad (13)$$

其中， $I(u)$ 表示需要重新审视的认知。这些规则使用可测执行信号实现该目标，而不依赖判断失败原因的预言机。每个步骤读取

$$z = (t_a, \Delta A, \Delta E, \text{lost}, \text{mismatch}, \text{first}, u_{\text{last}}, n_r).$$

其中， t_a 统计本次尝试中的轮次， ΔA 是新探索面积， $\Delta E = \lfloor \Delta n_{\text{frames}}/4 \rfloor + \Delta n_{\text{markers}}$ 是自上一检查点以来的观测数量代理。其余信号分别表示跟踪丢失、报告的楼层不匹配、首次访问、上一次恢复范围及已使用的恢复次数。进展定义为 $\text{prog}(z) = [\Delta A \geq 1.5 \text{ m}^2] \vee [\Delta E \geq 1]$ 。新帧数量不一定意味着新的场景内容，因此该测试作为防止过早恢复的实用保护条件。

令 f 表示已达到启用的回退阈值 T_f ， N 表示任务恢复次数上限。给定提议 $\hat{u}$ ，控制器按顺序应用以下规则。继续判定会立即返回。

$$\begin{aligned} u &= \text{continue} && \text{if no proposal and } \neg f, \text{ or } n_r \geq N, \\ u &= \text{continue} && \text{if proposed, } \neg f, \text{ and } t_a < t_{\min}, \\ u &= \text{continue} && \text{if proposed, } \neg f, \text{ prog}(z) \wedge \neg \text{lost} \wedge \neg \text{mismatch}, \\ u &= \hat{u} \text{ (proposed) or attempt (fallback),} \\ u &\leftarrow \max(u, \text{place}) && \text{if lost,} \\ u &\leftarrow \max(u, \text{floor}) && \text{if mismatch,} \\ u &\leftarrow \text{room} && \text{if } u = \text{house and first,} \\ u &\leftarrow u_{\text{last}}^+ && \text{if escalation is enabled, } n_r > 0, u \leq u_{\text{last}}, \\ u &\leftarrow \min(u, u_{\max}). \end{aligned} \quad (14)$$

配置提供 T_f 、 $t_{\min}$ 、 N 及最大范围 $u_{\max}$ 。记号 u^+ 表示比 u 高一级。提议置信度会被记录，但不作为这些规则的阈值。当 $N = 1$ 时，重复恢复升级不启用。楼层与房屋范围使用相同的保留地图和地点图简报；二者区别在于恢复提示要求重新评估的广度，而非不同的物理重置。

	上下文	位姿认知	路线认知	楼层认知	地图	标记	台账	存档
继续	保留	保留	保留	保留	保留	保留	保留	保留
尝试	替换	保留	保留	保留	保留	保留	保留	保留
地点	替换	重估	保留	保留	保留	保留	保留	保留
房间	替换	重估	重估	保留	保留	保留	保留	保留
楼层	替换	重估	重估	重估	保留	保留	保留	保留
房屋	替换	重估	重估	重估	保留	保留	保留	保留

所有恢复范围均保留物理轨迹与累计任务预算。

Figure 7 恢复范围与保留状态。 范围更大的恢复会重新审视更多认知，同时保留累计观测和任务记录。

会话被替换前，会获得最后一个仅含文件工具的步骤，并在分配的恢复笔记路径写入 JSON 交接记录：

$$\lambda_{k,a} = \text{WriteNote}_\theta(C_{k,a}) = (S, R, a^*, U, \tilde{U}, \hat{u}), \quad (15)$$

其内容涵盖已搜索地点、有证据支持的排除项、最后一个可靠锚点、未尝试选项、不确定性及建议范围。提示明确区分已搜索与已排除，并要求智能体指出支持每项排除的观测。因此，执行该尝试的会话会选择后继会话应知的信息。这些是执行者有证据支持的解释，而非经独立验证的排除项。恢复提示不鼓励重复这些搜索，因此错误排除也可能被带入下一次尝试。

编排器应用选定范围并打开新对话：

$$C_{k,a+1,0} = \text{Open}(\sigma, I_{\text{recover}}(g_k, \text{ref}(\lambda_{k,a}))). \quad (16)$$

Table 7 恢复范围。所有范围均保留地图、标记、台账与存档。

范围	重新审视的状态
继续 (<i>continue</i>)	不重置
尝试 (<i>attempt</i>)	活动对话
地点 (<i>place</i>)	对话与当前位置
房间 (<i>room</i>)	对话、位置与路线
楼层 (<i>floor</i>)	对话、位置、路线与楼层关联
房屋 (<i>house</i>)	与楼层范围相同的认知，并进行房屋层级重新评估

恢复指令给出范围、位置简报（如有）以及应首先读取的笔记。地图和其他文件仍可通过常规工具访问，机器人则保留同一目标、位置和剩余任务预算。重新审视位姿或路线认知不会删除已存储地图，也不会重置物理轨迹。

对于高于尝试的范围，具身执行框架会收集四幅周围视图。存在已存储的地点照片时，独立唤醒会话将其与当前视图比较并返回位置简报。楼层级和房屋级恢复还会提供命名地点图。这些观测帮助新尝试确定位置，而度量跟踪仍由定位工具负责。

重定位。任务间重定位不同于任务内恢复：环境已将本体移动到下一个起点。编排器保留空间状态与文件，收集四幅周围视图，并将其附加到下一任务的会话打开指令：

$$C_{k,1,0} = \text{Open}(\sigma, I(g_k, \rho_k) \parallel I_{\text{relocated}} \parallel F_k^{\odot}). \quad (17)$$

该推理会话先确定自身所在房间，再根据保留地图进行规划。此边界无须独立的房间识别会话。视觉识别与跟踪器的度量重定位仍是不同过程，因为识别房间本身无法确定机器人位姿。

整合。运行闭合时，执行会话结束，仅文件整合会话依据式 11 更新房屋笔记。其阅读列表包含台账、任务交接记录与先前房屋笔记；写入权限仅覆盖四个房屋笔记文件，这些文件随后可供后续运行使用。

A.6 验证

验证是完成检查机制，在闭合时承担两项职责（图 8）。停止前验证可将证据返回仍在运行的执行者；停止后认证决定存入记忆的状态。二者使用相同的证据接口，并默认使用与执行者相同的模型，但都不会接收执行者的对话或闭合时的自我评估。

证据与判定。调用 `close_task` 时，在任一检查运行前，会在建议停止点捕获四幅由程序选定的视图。判定器接收目标、这些视图、先前帧样本、地图，以及包含任务位置、已用步数、步数预算、帧数和标记名称的机械记录。帧选择器在十二帧限额内优先选择停止视图，其余位置从其他帧中均匀采样，并按时间顺序呈现结果。将该选择器记为 $\mathcal{S}_{12}$ ，检查形式为

$$v_k^{\text{pre}} = V_{\theta}(g_k, \mathcal{S}_{12}(F_k^{\text{pre}}), M_k^{\text{pre}}, m_k^{\text{pre}}), \quad v_k = V_{\theta}(g_k, \mathcal{S}_{12}(F_k), M_k, m_k). \quad (18)$$

上标标识首次受检查请求时可用的证据，第二个表达式则使用最终任务证据。二者均包含工具已捕获的停止视图。

判定为完成、未完成或未知，并附带证据陈述、已搜索地点名称与置信度。判定器没有动作工具。若响应无法解析，则在同一检查对话中提示一次，要求仅返回 JSON 判定。检查失败或未解决的响应记为未知。运行记录器在将判定与任务成功进行交叉制表时，会将未知与未成分开保留。

停止前验证。若交接文件缺失，编排器会先提示执行者写入一次。该文件检查独立于验证。对于首个到达判定器的请求，停止前验证仅在以下条件成立时拒绝 STOP：

$$v_k^{\text{pre}} \in \{\text{complete}, \text{incomplete}\} \quad \text{and} \quad v_k^{\text{pre}} \neq c_k.$$

验证将证据返回现有执行会话，使其能够再次观测或调整搜索。未知判定不会阻止闭合，后续请求会获准且不再否决。已检查请求计数器属于任务，在恢复后继续保留，仅在下一任务重置。一次性的交接记录缺失提示不会消耗本次检查机会。只有编排器会发出最终 **STOP**。这种有界检查既让执行者有机会响应，又不允许判定器无限期阻止闭合。

停止后认证。认证依据式 8 分配台账状态，但不改变已完成任务的分数。若基于模型的停止前验证和认证均已启用，且自停止前验证以来动作步数未变，则复用已存储判定；否则，认证会针对最终证据打开全新检查会话。未知虽然允许当前任务闭合，但在记忆中仍记为 **UNRESOLVED**。

两项检查使用独立开关，因为只有停止前验证能够改变当前展开。共享模型参数并不会使执行者与判定器的错误相互独立；将二者分开，是为了控制判定器可见的证据以及执行者可写入的记录。环境分数进入评测日志，而不会提供给基于模型的判定器或面向智能体的台账。第 4.3 节中的单视图干预改变了提供给判定器的视觉证据。

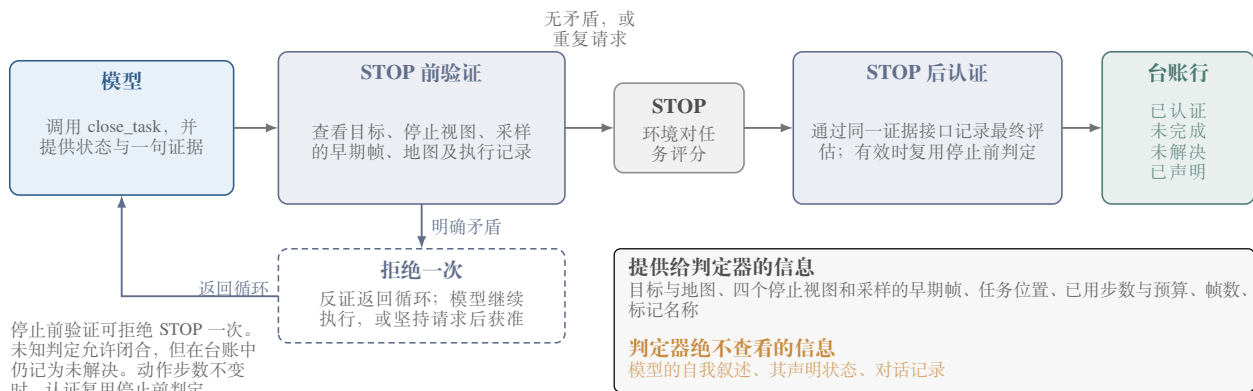


Figure 8 闭合时的验证。停止前验证可返回反证；认证随后记录结果，并在动作步数未变时复用停止前判定。

B 基准与评测协议

表 8 汇总本文在验证集未见划分上的评测。GOAT-Bench 上四种执行器的主要比较，以及 Opus 5 的会话与组件研究均使用完整验证集未见划分，其中包含 36 个场景、360 个回合中的 2,669 个子任务。

Table 8 验证集未见数据上的评测范围。

基准	场景数	回合 / 游程数	任务数	每个序列的任务数
GOAT-Bench	36	360 个回合	2,669	5–10
IR2R-CE	11	36 个游程	1,824	3–100

B.1 GOAT-Bench

GOAT-Bench 在 Habitat (Savva et al., 2019) 中基于 HM3D 扫描 (Ramakrishnan et al., 2021) 评估终身多模态导航。每个回合将智能体置于一栋房屋中，并依次给出一系列待到达目标；目标之间不会重置智能体。每个目标以三种方式之一指定：对象类别、指向某个实例的自然语言描述，或该实例的图像。当智能体在目标实例某个可导航视点的成功半径内停止时，视为到达目标；路径效率通过相对最短路径的 SPL 评分。本文评估 36 个场景中的全部 360 个验证集未见回合，保留所有 2,669 个子任务。在完整划分比较中，表 1 的各执行器及其独立会话基线使用相同评测任务，表 3 的 Opus 5 研究所有实验分支同样如此。每个 GOAT-Bench 回合和 IR2R-CE 游程都是一次独立基准运行，均从空地图、空任务记录和空房屋笔记开始。记忆会在同一次运行的任务间延续，但不会传给另一次运行，即使二者位于同一场景。因此，运行结束笔记不会为这些评测提供输入。在本文协议中，智能体位置、地图和标记均来自导航器自身的视觉 SLAM，而非模拟器；模拟器真实信息仅用于评分。

B.2 IR2R-CE

本文评估标准 IR2R-CE 验证集未见划分中的全部 36 个游程，保留其 11 个场景中的全部 1,824 项任务。IR2R-CE 基于 VLN-CE (Krantz et al., 2020) 构建，在 Matterport3D (Chang et al., 2017) 中评估游程级指令跟随。一个游程在连续环境中连接同一场景内连续的 Room-to-Room 回合；即使智能体被转移至下一回合起点，仍会将前一回合学到的内容延续至下一回合。每个回合按 R2R-CE 使用成功率和 SPL 评分；整个游程则使用 t-nDTW 评分，即游程实际行驶路径与参考路径之间的归一化动态时间规整。与 GOAT-Bench 相同，本文使用验证集未见划分。t-nDTW 采用基准的标准轨迹处理与评分程序计算，并遵循与比较方法相同的约定。

B.3 比较来源与设置

表 1 的“基础模型”列标明所报告导航结果使用的主要语言模型或视觉语言模型，但未列出所有感知或建图模块。例如，ObsGraph 还使用 GPT-4o-mini 构建房间层次结构 (Lee et al., 2026)，AstraNav-Memory 则以 DINOv3 视觉编码器增强 Qwen2.5-VL-3B (Ren et al., 2025)。已发表的 3D-Mem GOAT 结果使用 GPT-4o 从记忆和前沿快照中选择目标，移动由 Habitat 路径查找器处理 (Yang et al., 2025)。GSMem 也使用 GPT-4o 执行 GOAT 推理，但还结合高斯场景重建、检索渲染和探索规划 (Lu et al., 2026)。即使推理模型名称相同，这些感知与控制流程仍不一致。本文的相同骨干模型对照保持执行器和导航接口不变，仅改变外围具身执行框架；其增益衡量包括记忆、恢复和完成检查在内的完整系统变化。

主表中的既有方法沿用原始评测协议。表 1 的剑号标出采用 278 子任务评测的方法，其中包括本文的 GPT-4o 子集结果。本文未标记的行和相同骨干模型增益均使用完整划分。本文报告 3D-Mem 的子集结果 69.1 s-SR 和 48.9 SPL，而非其完整集合结果 62.9 和 44.7 (Yang et al., 2025)。MemoryExplorer 报告相同的 36 场景、每场景一个回合协议 (Wang et al., 2026c)。GSMem 所述验证集未见基准每个场景含十个回合 (Lu et al., 2026)。表 1 和表 2 的 GT 位姿列标明使用模拟器所提供定位的方法。CMA、TourCMA、PoolCMA 和 PoolEndCMA 策略则使用 RGB-D、指令和动作历史，不输入位姿 (Krantz et al., 2023)。本文导航器使用视觉 SLAM 位姿，模拟器真实信息仅用于评测。表 9 保留了因篇幅限制未纳入主表的其他已发表基线。

Table 9 其他已发表的 GOAT-Bench 验证集未见结果。所列方法均使用模拟器提供的位姿。

方法	类型	s-SR	SPL
SenseAct-NN Monolithic (Khanna et al., 2024)	训练式	12.3	6.8
Modular CLIP on Wheels (Gadre et al., 2023; Khanna et al., 2024)	零样本	16.1	10.4
VLMnav (Goetting et al., 2024; Ji et al., 2025)	零样本	20.1	9.6
Modular GOAT (Khanna et al., 2024)	零样本	24.9	17.2
DyNaVLM (Ji et al., 2025)	零样本	25.5	10.2
SenseAct-NN Skill Chain (Khanna et al., 2024)	训练式	29.5	11.3
TANGO (Ziliotto et al., 2025)	零样本	32.1	16.5
AnyGoal, 单智能体 (James et al., 2026)	零样本	41.9	14.4
MTU3D (Zhu et al., 2025)	训练式	47.2	27.7

GOAT e-SR 遵循 Niu et al. (2026) 的全子任务判据。原始 Monolithic 基线报告 12.3 s-SR 和 6.8 SPL，但未报告 e-SR。Niu et al. (2026) 报告的 0.8 e-SR 属于其重新实现，对应 12.6 s-SR 和 8.8 SPL，未与原始结果行混合。对于 IR2R-CE，CMA 变体和 MAP-CMA 来自 Krantz et al. (2023)；OVER-NAV 使用其连续环境结果 (Zhao et al., 2024)。ETPNav 和 HNR 使用 Han et al. (2026) 表 2 中的 IR2R-CE 评测结果，而非其原始单回合 R2R-CE 分数。SeqWalker 同样使用该表，而非其另一个 SH IR2R-CE 基准。表 10 保留了因篇幅限制未纳入主表的三个 CMA 变体。

Table 10 其他已发表的 IR2R-CE 验证集未见结果 (Krantz et al., 2023)。

方法	类型	GT 位姿	s-SR	SPL	e-SR	t-nDTW
TourCMA	训练式	×	18	17	—	36
PoolCMA	训练式	×	16	15	—	36
PoolEndCMA	训练式	×	18	16	—	38

本文使用 Qwen3.8-27B、GPT-4o、Opus 5 和 GPT-6 Astra 评估 NavHarness。四种执行器均使用相同的建图、记忆与恢复机制。遵循 MIP (Zhou et al., 2026)，推理核心采用多轮编程智能体会话。GPT-6 Astra 使用中等推理强度的 Codex，Opus 5 使用 Claude Code，Qwen3.8-27B 和 GPT-4o 均使用 mini-SWE-agent。NavHarness 管理这些会话外围的任务与恢复边界，而非替换其内部推理循环。所有受控组件比较均使用 Opus 5。

辅助角色的模型分配。标准启动器将所有角色路由至同一模型。停止前验证和停止后认证使用相同判断器模型。执行会话在被替换前写入恢复交接记录；启用整合时，由使用相同模型的独立会话完成整合。除附录 B.7 的判断器互换比较外，导航及所有辅助角色均由同一模型承担，从而在每项相同骨干模型比较中固定模型分配。这些角色均不接收评测器成功标签。保存的 Opus 和 Astra GOAT 运行结果确认使用同模型判断。独立会话基线不使用判断器或跨任务整合。

默认停止程序。基准配置检查首次停止请求并认证已完成任务 (precheck=model、certify=model)。表 3 中“无停止后认证”仅移除停止后裁决，停止程序保持不变。独立会话基线省略两项检查；额外的“独立会话 + 验证”对照启用停止前验证。完整系统的停止前验证消融使用 precheck=none，同时保留 certify=model。

两种基线会话。独立会话为每项任务启动新的编程会话，仅配备观测与动作工具。目标、观测、推理与工具响应均保留在当前对话中。会话无法访问外部工作记忆或长期记忆，包括地图、标记、台账条目或交接记录文件。单一会话将同一对话从一项任务延续至下一项任务，并在上下文窗口填满时执行后端压缩 (附录 A.3)。会话基线与完整具身执行框架使用相同任务预算，比较的是完整系统策略。参考配置使用全文台账访问 (ledger=text)。

方法标签。主表中的“类型”标签是描述性的，并非互斥的方法类别。自主式表示由工具介导的模型推理，训练式和零样本描述是否采用任务专用训练，混合式表示将学习式导航与高层推理结合。一个系统可能同时符合多种描述；这些标签并不意味着骨干模型等价。

任务预算。表 11 给出 GOAT 受控比较使用的统一导航预算。恢复与原任务共享剩余配额。运动接口见附录 A.2，会话管理见附录 A.3。

Table 11 GOAT 受控比较的预算核算。

项目	限制或计数规则
移动预算	每项任务最多执行 500 个前进或转向动作
模型轮次预算	每项任务最多 200 个导航模型轮次
动作批处理	每个已执行动作单独计费
恢复配额	最多恢复一次，并共享任务剩余预算
后备触发	80 轮，不重置任一预算
观测与记忆	不计入移动动作

运行时间与上下文占用。在 GOAT-Bench 验证集未见划分的性能分析运行中，Opus 5 和 GPT-6 Astra 每项任务的摊销挂钟时间分别为 129.5 s 和 67.8 s。我们将被评测场景运行的耗时求和，再除以任务数，其中包含设置、导航和边界处理。二者每项任务的平均峰值提示规模分别为 20.7k 和 23.2k 个词元，用于描述上下文占用。端到端挂钟时间包含已记录的辅助处理。附录 A.3 中的输入成本示例基于给定用量和价格假设，而非实测 API 账单。

B.4 统计方法

四种执行器在 GOAT-Bench 上的完整具身执行框架与独立会话比较，以及 Opus 5 的会话和组件研究，均在 36 个场景、360 个验证集未见回合中的相同 2,669 个子任务上使用三个随机种子。表 1 和表 3 报告各次运行分数的均值与标准差。IR2R-CE 上 NavHarness 和独立会话的结果同样报告四种执行器在三个随机种子上的均值与标准差 (表 2)。Opus 的 GOAT-Bench 比较按场景和回合标识符对共享任务进行配对，每项任务在每次运行中计数一次。配对 95% 置信区间汇总重复运行，并执行 5,000 次按场景聚类的自助重采样，同时保持配对结果不分离。这些是组件分析的逐点区间，未进行多重比较校正。对于持续部署 (图 3)，分数和配对变化汇总全部 36 栋房屋的任务。95% 置信区间以房屋为重采样单位，并保持“完整系统”与“无整合”的结果配对。

B.5 链式成功

链式成功是序列中每项任务均成功的序列比例：在 GOAT-Bench 上以回合为序列，在 IR2R-CE 上以游程为序列。该指标反映部署真正关心、但任务成功率会掩盖的情况，因为序列中任意一次失败都会导致整条链失败。

B.6 干预设置

表 3 将系统比较 (A) 与组件干预 (B–D) 分开。所有实验均使用 Opus 5，并采用顺序相同的完整划分 2,669 子任务评测。参考配置保留地图与任务记录，使用结构化恢复交接记录，并启用停止前验证与基于四个停止视图的停止后认证。B–D 的每一行均独立从该参考配置开始，不继承上一行的改动。所有差异和置信区间始终相对于完整系统，包括“独立会话 + 验证”这一行。

A. 完整会话策略。独立会话为每项任务启动仅含上下文的编程会话，即使在任务内部也不使用外部记忆。单一会话按照附录 A.3 的定义，使同一对话跨任务延续。“独立会话 + 停止前验证”在不增加记忆的情况下为该策略加入停止检查。将这一对照与完整系统比较，可在保持停止前验证启用的同时联合检验其余具身执行框架特性；上述各行都不是纯记忆消融。比较 62.4 与 58.9 所描述的对照不同于表 3 中的配对变化，后者比较 62.4 与完整系统的 81.5。

B. 跨任务记忆。“每项任务后清空地图（保留记录）”会移除地图和空间标记，但保留任务记录及每项任务内的建图能力。不同于仅含上下文的独立会话，该对照仍使用工作记忆。“隐藏早期任务记录（保留地图）”会保留地图，但不提供早期任务的文本历史，包括台账、先前交接记录和历史任务文件。这些记录既不会出现在初始上下文中，也无法通过文件工具访问，但仍可为评测而保存。当前任务可写入自身记录，并在该任务的不同尝试间传递恢复记录。这些干预检验空间知识与早期尝试记录能否相互替代。恢复、结构化交接记录和两项验证功能均保持启用。额外的“无跨任务记忆”对照将上述两项干预结合。每项任务开始时都不继承地图、空间标记、台账条目或交接记录，但可以构建和查询自己的地图与记录，包括在该任务各次尝试间传递的恢复记录。任务预算和两项完成检查保持不变。其标准基准结果见表 3。第 4.4 节的扩展部署采用另一种干预：“无整合”按照与完整系统相同的持久化规则保留地图与任务记录，但禁用运行结束时写入房屋笔记的会话。二者使用相同执行器、任务序列、预算、恢复和完成检查。该比较衡量整合后的长期记忆在已保留空间记录和任务级记录之外的附加价值。

C. 恢复可用性与交接内容。“无恢复”同时移除请求式和后备式重新启动，但保留完整的 200 轮与 500 步任务预算。三项交接记录干预保持恢复触发条件、作用域控制和共享预算不变。空交接记录不提供失败尝试的记录；其他任务的交接记录提供错配记录；等长摘要则以普通摘要替代结构化记录。第一项对照检验恢复是否有益，其余对照检验哪些信息必须跨重新启动传递。它们并未关闭恢复，而摘要比较同时检验信息选择与组织方式。

D. 验证用途与证据。“无停止前验证”保留停止后认证，但绝不拒绝 STOP。“无停止后认证”保留停止前验证，但任务记述中不包含停止后的认证状态。“仅一个停止视图而非四个”保留两项验证功能，只将停止位置的证据从环视改为前向帧。因此，它检验的是提供给验证的证据，而非任一检查是否存在。这些是相互独立的运行轨迹，改变某个机制可能会改变后续观测、轨迹和记忆内容。

B.7 记忆访问与判断器选择

固定访问历史记忆。为检验在保留记录之外，选择检索时机与检索内容是否重要，我们以固定访问策略替代智能体主导的检索。建图工具、记录写入、恢复交接记录和完成检查保持不变。两种条件均使用 Opus 5，接收相同初始记忆条目，并可自由查询在线地图。对照策略在任务开始时、恢复后以及每十个导航模型轮次检索一次历史文本。固定查询将目标描述与最新观测描述拼接起来，两种条件均使用相同描述接口。BM25 在 2,000 词元预算内最多检索五个原始片段，其中包含来源和记录状态。记录按 400 词元分块，相邻块重叠 50 个词元；重叠片段会被合并，得分相同时优先选择较新的记录。导航器仍可查阅当前任务记录并查询地图，但无法通过其他文件工具读取历史文本以绕过固定策略。

本文使用与表 3 相同的 GOAT-Bench 完整划分评测、三次运行协议和任务预算。相较固定访问，智能体主导访问使 s-SR 提高 7.0 个百分点 (95% 置信区间 [+5.9, +8.2])，SPL 从 49.0 提升至 55.0 (表 12)。因此，按固定时间表提供原始记录无法完全获得让导航器随观测到来而自行选择查询的全部收益。该比较联合检验查询

时机、内容选择与后续跟进。两种条件均从空记忆开始，以相同方式保留记录，但随着轨迹分歧，其后续内容可能不同；二者均不继承早期运行的房屋笔记。

Table 12 使用 Opus 5 在 GOAT-Bench 上比较智能体主导与固定访问的历史记忆。分数为三次运行的均值 $\pm$ 标准差。配对差异为 NavHarness 减去固定访问。参考挂钟时间来自附录 B.3 的性能分析运行。

指标	NavHarness	固定访问
s-SR (%)	81.5 $\pm$ 0.7	74.5 $\pm$ 0.8
SPL	55.0 $\pm$ 0.6	49.0 $\pm$ 0.7
平均时间 / 任务 (s)	129.5	137.2
配对 Δ s-SR (百分点)	+7.0	
95% 置信区间 (百分点)	[+5.9, +8.2]	

Table 13 同模型与跨模型判断。两项完成检查均使用所列判断器，其余角色保持不变。分数为三次运行的均值 $\pm$ 标准差。

导航器	判断器	s-SR (%)	SPL
Qwen3.8-27B	Qwen3.8-27B	71.7 $\pm$ 0.5	48.2 $\pm$ 0.4
Qwen3.8-27B	Opus 5	71.3 $\pm$ 0.4	48.5 $\pm$ 0.5
Opus 5	Opus 5	81.5 $\pm$ 0.7	55.0 $\pm$ 0.6
Opus 5	Qwen3.8-27B	81.7 $\pm$ 0.6	55.2 $\pm$ 0.5

跨模型判断。我们互换 Qwen3.8-27B 与 Opus 5 的判断器，以检验导航性能是否依赖使用导航器自身模型执行完成检查。每个导航器分别使用默认判断器与另一模型进行评估，后者同时负责停止前验证和停止后认证。导航器、恢复记录写入器和整合模型保持不变，各配置所获证据、指令、停止规则与任务预算也保持不变。

判断器互换使 Qwen 的 s-SR 变化 -0.4 个百分点，Opus 变化 $+0.2$ 个百分点，SPL 变化同样很小（表 13）。因此，在两种配置中，使用不同判断器时的导航性能都接近默认设置。这并不表示完成检查不重要：表 3 D 组单独检验了其贡献。相近的导航分数也不能证明判断准确率相同，亦不能排除两种模型共有的错误。

C 完整结果

本附录给出主表所依据的配对统计数据，以及各组件的诊断结果。GOAT-Bench 分数比较使用验证集未见划分 360 个回合中的全部 2,669 个子任务，与表 1 和表 3 一致。文中明确标注了三随机种子分数估计和置信区间。下述事件百分比为描述性日志汇总，不应理解为额外的三随机种子效应估计。

C.1 组件效应概况

图 9 汇总表 3 B–D 组中对 Opus 的受控干预，包括等长普通摘要对照。停止前验证干预与认证分开列示。图中以配对不确定性展示各项干预的总体效应。下述诊断将这些效应与重复探索、失败尝试和不准确的完成记录联系起来。表 3 同时报告分数和配对置信区间。每个配置的区域汇总三个随机种子，采用附录 B.4 中按场景聚类的程序。

C.2 会话与系统比较

独立会话仅使用当前任务的对话，不使用外部地图、任务记录或长期记忆；这不同于保留任务内建图的地图重置消融。表 1–3 报告系统和会话策略比较，其中包括四种推理模型在 IR2R-CE 上的独立会话结果。

在完整的 2,669 子任务评测中固定 Qwen3.8-27B 时，NavHarness 将 s-SR、SPL 和 e-SR 分别提高 30.3、20.7 和 11.3 个百分点。使用 GPT-4o 时，对应增益为 34.8、26.9 和 20.5 个百分点。在同一评测中固定 GPT-6 Astra 时，NavHarness 相较独立会话将 s-SR、SPL 和 e-SR 分别提高 18.6、22.2 和 23.3 个百分点。这些数值为总体差异。配对不确定性需要相应的任务级记录，下述配对区间仅针对 Opus。

在完整划分的 Opus 研究中，相较完整系统，独立会话的成功率下降 22.6 个百分点（95% 置信区间 $[-26.3, -19.0]$ ），单一会话下降 26.6 个百分点（ $[-30.5, -22.8]$ ）；二者的 SPL 分别降低 20.1 和 22.2 个百

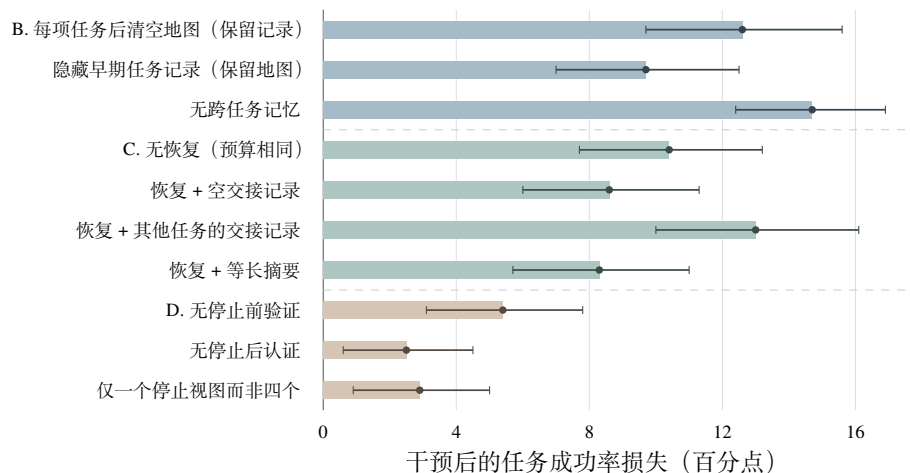


Figure 9 使用 Opus 5 在 GOAT-Bench 完整划分上的组件效应，分组方式与表 3 一致。条形表示相较完整具身执行框架的损失，误差线为三个随机种子上配对变化的 95% 置信区间。正值表示完整具身执行框架更优；所有区间均不包含零。

分点。这些比较评估完整系统策略。SPL 同时结合任务成功率与路径效率，因此上述总体差异无法分离两个系统共同完成任务上的路径节省。表 3 给出所有组件分数和配对成功率区间，包括空记录和普通摘要对照。移除持久地图和台账还会使 SPL 分别降低 15.2 和 13.2 个百分点。

Table 14 GOAT-Bench 上按目标模态划分的完整具身执行框架结果。数值为百分比。

目标	s-SR	SPL
对象类别	86.5	62.4
语言描述	75.2	48.8
图像	82.2	53.0

C.3 任务内恢复与交接记录

等长普通摘要对照达到 73.2% s-SR 和 47.0% SPL，而完整具身执行框架分别为 81.5% 和 55.0%。其 s-SR 配对变化为 -8.3 个百分点，95% 置信区间为 [-11.0, -5.7] 个百分点。该比较检验结构化记录相对于一般摘要的效果，并补充表 3 中空记录和任务错配记录干预。

图 10 从三个角度分解 GOAT-Bench 上的恢复事件。完整具身执行框架约在四分之一的任务上进行恢复；其中 39% 由模型请求，61% 由 80 轮后备规则触发。在各实验分支记录的恢复事件中，使用本次尝试自身记录时有 56.1% 最终成功，空记录为 31.7%，不同任务的记录为 35.0%。模型请求发生得更早，中位数为 32 轮，后备触发则为 84 轮；模型请求的成功率也更高，为 61.5%，后备触发为 52.5%，但两种触发方式选中的失败类型不同。不同记录干预下，进入恢复的任务集合也可能不同。这些条件比率描述各次运行中观测到的事件，表 3 则检验记录干预对总体任务成功率的影响。

C.4 跨任务记忆

禁用所有跨任务记忆，同时保留任务内建图、恢复和验证时，s-SR 为 66.8%，SPL 为 37.4%。s-SR 的配对变化为 -14.7 个百分点（95% 置信区间 [-16.9, -12.4]），从其他部分保持完整的具身执行框架中分离出任务间延续记忆的贡献。移除任务记录或空间持久性都会增加实际探索。完整具身执行框架下每项任务平均 127 步；无台账时增至 159 步，无持久地图时增至 168 步。无地图时，模型每项任务放置的标记数从 1.3 增至 3.4（约为 2.6×），请求恢复的频率增至 1.8 倍，每项任务都要重建上一任务已记录的信息。

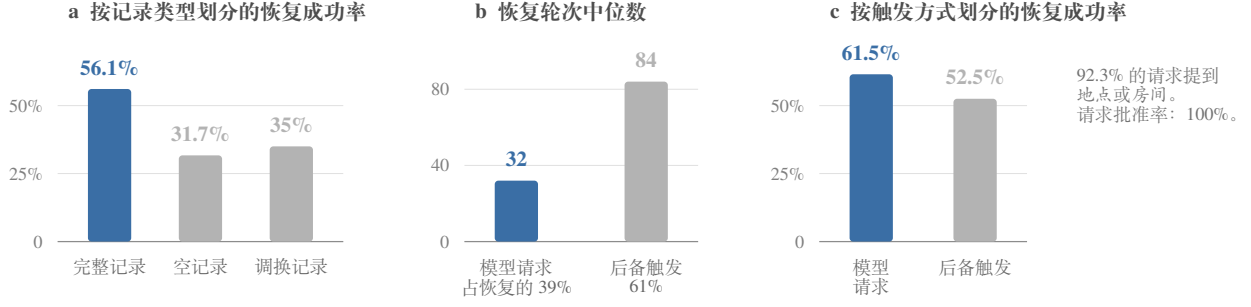


Figure 10 GOAT-Bench 上的恢复。左：按传递给新尝试的内容划分，最终成功的恢复比例。中、右：模型请求恢复远早于后备触发，且其请求成功率更高。

C.5 验证用途与证据

停止前验证干预。仅禁用停止前验证会使完整系统的成功率从 81.5% 降至 76.1%，变化为 -5.4 个百分点，配对 95% 置信区间为 $[-7.8, -3.1]$ ；SPL 从 55.0% 降至 47.0%。三个随机种子上的区间不包含零。当两个配置均启用停止前验证时，完整系统的成功率比独立会话高 19.1 个百分点（95% 置信区间 16.4–22.4）。该比较固定停止检查，同时联合评估其余具身执行框架特性，包括持久状态与恢复。

日志中的局部停止效应。首次被拒绝的停止请求为停止前验证提供了另一种观察视角。对于任务 i ，令 y_i^{stop} 表示在该记录状态执行 STOP 时的评测成功值， y_i^{final} 表示请求被拒后任务的最终成功值。一次挽救对应 $(0, 1)$ ，一次损害对应 $(1, 0)$ 。局部净变化为

$$\hat{\delta}_{\text{stop}} = \frac{1}{N} \sum_{i \in \mathcal{R}} (y_i^{\text{final}} - y_i^{\text{stop}}), \quad (19)$$

其中， $\mathcal{R}$ 包含首次请求被拒的任务， N 是 GOAT-Bench 验证集未见划分诊断日志中的任务数。该配对比较以启用停止前验证的策略所产生的历史为条件，估计在这些状态继续执行对当前任务的影响，而非在不使用停止前验证时重新运行完整序列的效应。恢复 y_i^{stop} 需要使用该状态下的基准成功判据，而不是智能体声明或判断器裁决。

Table 15 停止前诊断。除以百分点表示的净变化外，其余数值均为百分比。

诊断项	完整系统	独立会话 + 停止前验证
首次停止请求被拒的任务	11	20
拒绝精确率	73	75
正确拒绝停止后的挽救率	45	35
挽救 / 全部任务	3.6	5.0
损害 / 全部任务	0.7	1.1
局部成功率净变化	+2.9	+4.0

这些汇总结果表明，独立会话更常到达存疑的停止状态，而两种配置的拒绝精确率相近。局部估计衡量 STOP 被拒后继续执行的效果，运行轨迹差异则包含整个轨迹与后续状态的变化。

停止后认证。在完成声明中，四视图认证判断器正确拒绝虚假声明的比例为 12.2%，错误拒绝真实声明的比例为 4.1%（表 16）。两个百分比均以所有完成声明为分母。下一项任务会同时接收裁决、声明和任务记述（图 11）。在虚假记录之后，有认证时后续任务成功率为 66.7%，无认证时为 61.4%；在真实记录之后，二者分别为 82.7% 和 81.6%。这些后续任务比率以各配置产生的历史为条件，而非匹配的前序状态，故无法分离认证效应。表 16 比较四个停止视图和单个前向帧下认证判断器的裁决与环境真实结果。两次运行的终止观测不同，因此该比较描述的是各次运行，而非单独分离视图数量的影响。

在同一完成声明总体上，四视图准确率为 90.7%，而接受所有声明时为 82.6%。单视图运行中的对应数值为 82.9% 和 80.6%。根据上方舍入后的百分比计算，科恩 κ (Cohen’s κ) 分别约为 0.67 和 0.40。四视图检查器拒

Table 16 停止后认证结果与准确率（占完成声明的百分比）。括号内为全盘接受的准确率。

证据	认证为真， 实际为真	认证为真， 实际为假	拒绝， 实际为真	拒绝， 实际为假	准确率 (全盘接受)	κ
四个停止视图	78.5	5.2	4.1	12.2	90.7 (82.6)	0.67
单个前向帧	74.2	10.7	6.4	8.7	82.9 (80.6)	0.40

绝 70.1% 的虚假声明，并接受 95.0% 的真实声明；在其所有拒绝中，25.2% 涉及真实完成。两个检查器的准确率均高于全盘接受，其中四视图运行的优势更大。这些结果支持使用认证区分有证据支持的结果与无证据支持的完成声明，同时也表明误接受和误拒绝仍可能发生。停止后认证消融保持停止前验证不变（附录 B），因此无法分离该检查对导航成功率的影响。

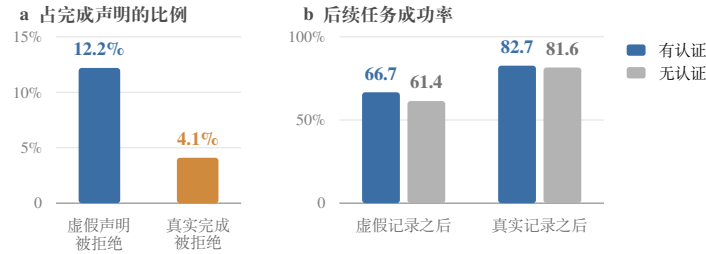


Figure 11 GOAT-Bench 上的验证。左：四视图拒绝占完成声明的比例。右：在台账中虚假或真实记录之后，有无认证时后续任务的成功率。

D 持续部署与机制分析

D.1 组件效应与故障诊断

表 17 将干预造成的总体损失与相应运行中观察到的故障模式联系起来。这些诊断有助于解释各组件的作用，但并非按难度或故障类型分层后得到的独立效应估计。

Table 17 组件效应及相关诊断。s-SR 的变化来自总体干预，而非在各压力条件下估计的效应。笔记干预使用另一任务的笔记进行替换。

组件	Δ s-SR	所吸收的问题	移除后的表现
地图和标记	-12.6	重访与位移	步数 +32%，写入标记约为 2.6×
台账	-9.7	重复目标	即使保留地图，步数仍 +25%
恢复笔记	-13.0	停滞的搜索	恢复成功率由 56% 降至 35%
恢复	-10.4	无效尝试	请求远早于兜底机制触发
四视图证据	-2.9	模糊的任务闭合	不同运行中的完成声明 κ : 0.40 对 0.67
认证	-2.5	缺乏依据的声明	12.2% 的错误完成与 4.1% 的真实完成被拒绝（占全部声明的比例）

D.2 多房屋场景下启用与关闭整合的对比

部署日程将 GOAT-Bench 扩展为同一房屋内的十次巡游，并分布在多个模拟日中。单次巡游内的任务保持 GOAT-Bench 的连续运动，但巡游边界并不连续：机器人会被放置在下一次巡游规定的起点。这模拟了家用机器人关机后恢复工作的情形。机器人必须根据当前观察、SLAM 和记忆中的地标重新确定方位，才能复用已存路线和目标位置。下面的空间记录案例考察 SLAM 坐标系变化时会话如何处理继承的坐标，以及描述性知识如何继续指导导航。

扩展评测覆盖全部 36 所房屋（图 3）。本附录详细分析其中三所房屋（记为 A、B、C），在匹配的任务序列上比较同一执行器的两种配置。两种配置均以相同的持久化与访问规则保留地图、标记、地点图和任务记录，包括台账与交接记录。无整合对照仅禁用运行结束时的整合；完整配置还会将日志提炼为房屋笔记，供后续运行中的导航会话通过文件工具读取。其余设置完全相同：Claude Opus 5、每个任务一个会话、每个任务 200 轮和 500 步、至多一次恢复、四视图认证判别器，以及停止前验证。每所房屋的第一天尚无笔记，执行器只能读取当天不断追加的日志。执行器可以读取笔记但不能写入：Read、Glob 和 Grep 仅可访问笔记目录，所有 Write 或 Edit 均会被拒绝。在案例日志中，*ledger* 或 *journal* 也指包含执行者叙述在内的完整任务记录，而不只指 run/ledger.md 中的紧凑索引。下文将这些详细摘录标为任务记录，引文保留原始措辞。方框中的段落均为模型原文，并用高亮标出：对自身记忆的判断、对错误或代价的承认、给未来自己的建议或诊断。

明确要求与未明确要求的行为。两种配置都能在导航期间构建工作地图、添加标记并生成任务记录，也都能在之后检索这些记录。只有完整配置会将经验整合为索引、房屋概览、房间笔记和导航技能。该对比检验这种长期组织方式相较于持久地图和任务记录带来的额外价值。

两种配置接收相同的部署任务提示、工具说明和交接指令。日志使用 Read、Glob、Grep 等 CLI 原生命名。表 18 说明案例记录对应的指令。两种配置都能访问台账和详细任务记录，只有完整配置能够获得整合产生的房屋笔记。

表 18 将明确要求的操作与执行期间观察到的更具体行为区分开来。未被显式编写进流程的行为，仍可能由一般记忆指令引出。

Table 18 部署指令与观测到的行为。未显式写入脚本的行为仍可能得到通用记忆指令支持。

行为	指令或工具支持	关系
报告目标、结果和经过的房间	交接记录指令	已请求
报告放置了哪些标记	交接记录指令	已请求
记录未成功的搜索	交接记录指令	已请求
移动标记	标记工具支持重新放置	工具支持
质疑早期任务记录	关于记录可能有误的警告	已请求
修正特定标记	维护标记的一般权限	未写入脚本
诊断特定仪器故障	通用推理与恢复工具	未写入脚本
拒绝已存路线，依靠视觉导航	访问记忆和当前视图	未写入脚本
根据当前视图核验笔记	将笔记视为回忆的指令	已请求
识别特定的过时笔记	关于笔记可能过时的一般警告	未写入脚本
尝试编辑继承的笔记	拒绝写入允许路径之外	未请求的尝试

每个案例房屋中，两组使用相同的任务顺序。图 15 给出各房屋的独立画像，图 12 汇总完整运行，图 3 报告 36 所房屋的汇总评测。

不同环境，不同增益。房屋 A 的成功率与路线效率均有改善，SPL 增益在翻转为成功的任务和两条运行均完成的任务之间大致均分。房屋 B 的成功率本已较高、差异很小，但路线效率显著提升：两条运行均完成的任务路径更短，十步内闭合的任务比例由 8% 升至 18%。房屋 C 的增益几乎全部来自成功率，两条运行均完成的任务看似没有效率提升；这是串联起点造成的伪象，因为在起点一致的子集上，各环境的效率都在提升（图 13）。按目标模态看，描述性目标的增益最广：房屋 A 的成功率从 77% 升至 95%，房屋 C 从 60% 升至 72%；每任务步数分别从 217 降至 93、从 204 降至 157。房屋 B 的物体目标均被两条运行完成，差异仅在步数，分别为每任务 66 步和 41 步。任务闭合判别器在无整合时错误接受完成的比例为 10%、1.5%、19%，启用整合时为 3%、0%、18%（房屋 A、B、C）；错误拒绝真实完成的比例分别为 8–11% 和 6–15%。以下 14 个案例从 48 个分析案例中选出，用于说明空间修订、失败记录、仪器诊断、整合与选择性复用；可用时还给出对应的无整合执行。这种按行为选择的案例同时包含收益与失败，不用于估计其在所有导航任务中的发生频率。案例追踪记忆如何进入具体决策。组件效应通过受控干预检验；完整系统与无整合轨迹之间的差异，不会进一步分解为各记录或工具的独立增益。

逐任务观察（图 15），房屋 A 的完整运行仅在第二天成功率低于无整合运行；此时同一张灯桌发生了两次失败（案例 7），但完整运行的步数始终更少。房屋 B 的成功率曲线接近，较大差异出现在前两天的步数：衣

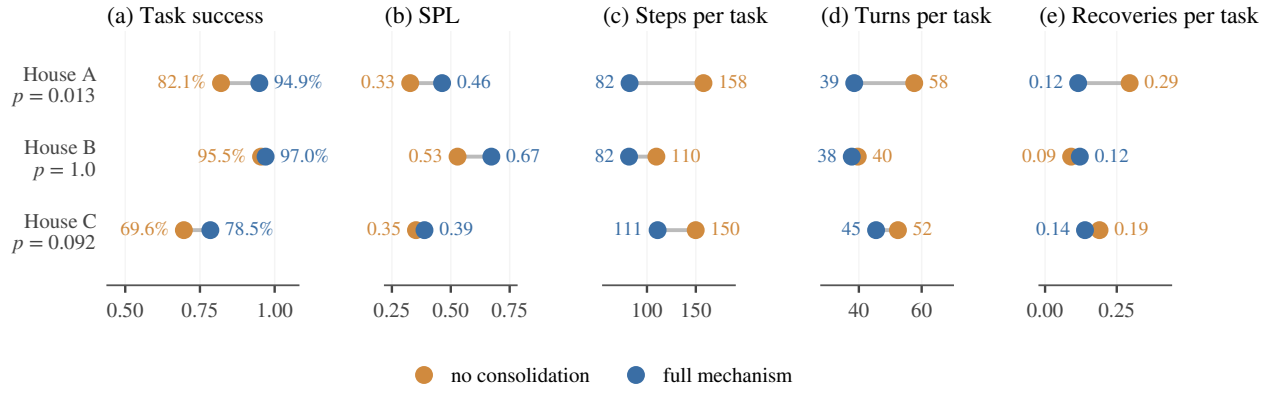


Figure 12 部署 A-C 的汇总对比。成功率和 SPL 使用环境真值；步数、轮数和恢复次数为每任务均值； p 为针对任务成功的精确 McNemar 检验。

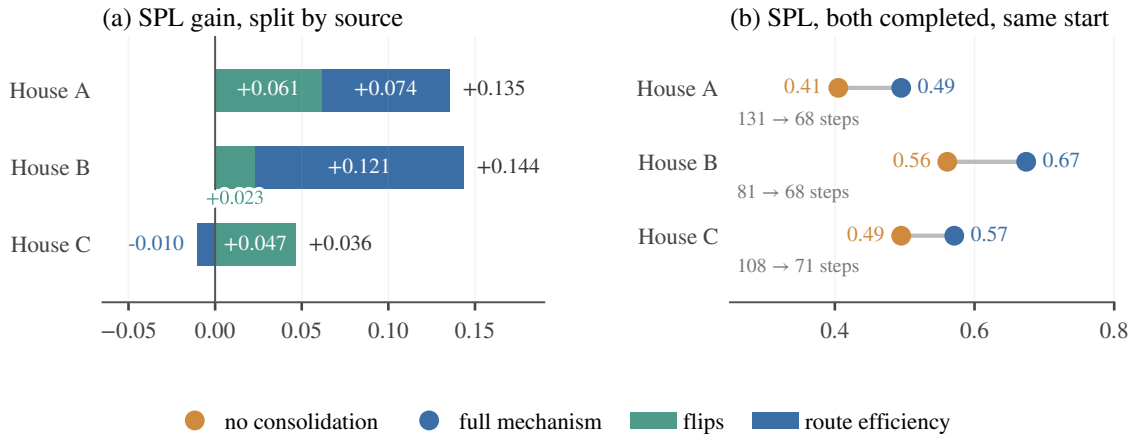


Figure 13 SPL 增益的来源。(a) 将增益分为仅一条运行完成的任务（翻转）与两条运行都完成的任务（路线效率）；房屋 C 的效率项为负。(b) 由于任务串联，两条运行并不总从同一地点开始任务，且 SPL 按各自最短路径归一化；在起点距离小于 0.5m 且均完成的任务上，每个环境的效率均有提升。

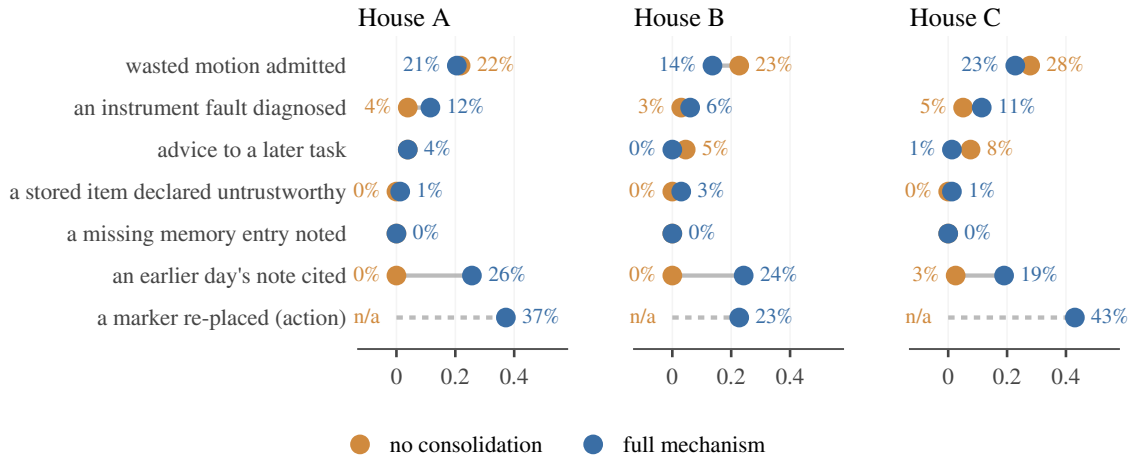


Figure 14 任务记录中的关键词匹配率。文本提及不一定对应经过验证的记忆读取或修订。最后一行表示动作而非短语：在已有标签下重新放置标记。两种配置均保留标记；未制表的比率记为 n/a。

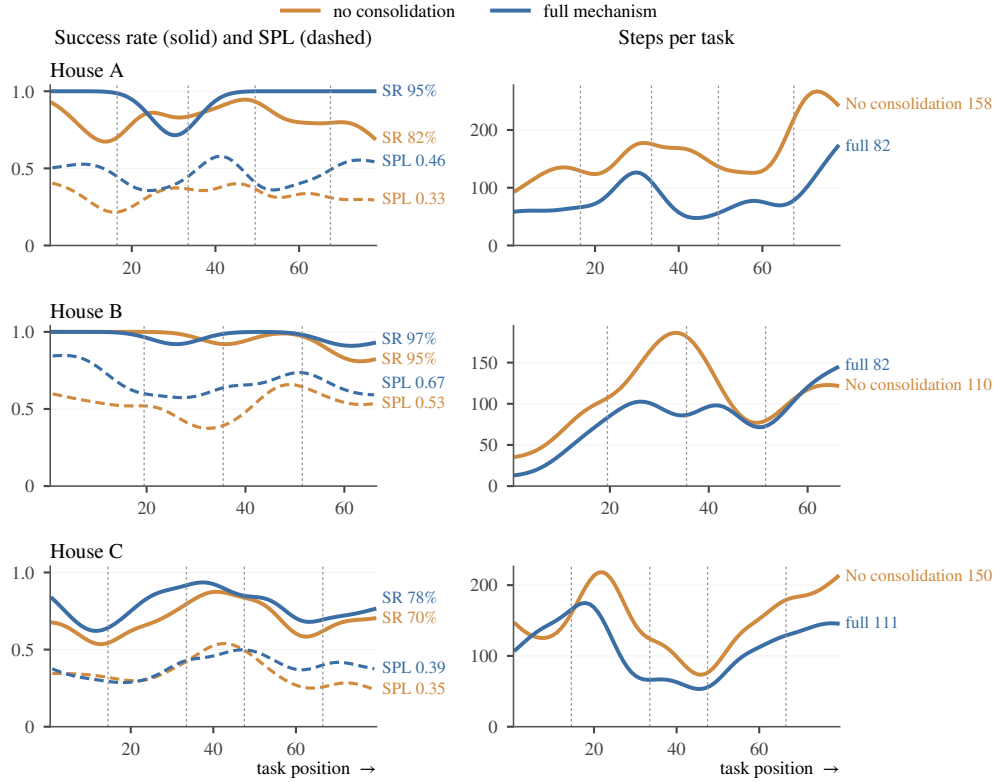


Figure 15 部署 A–C 的性能趋势。左图为成功率（实线）和 SPL（虚线），右图为每任务步数。赭色表示无整合，蓝色表示完整机制。每条曲线末端的标签是该运行在全部任务上的均值；竖直虚线是完整运行的日期边界。每个任务的原始结果见图 16。

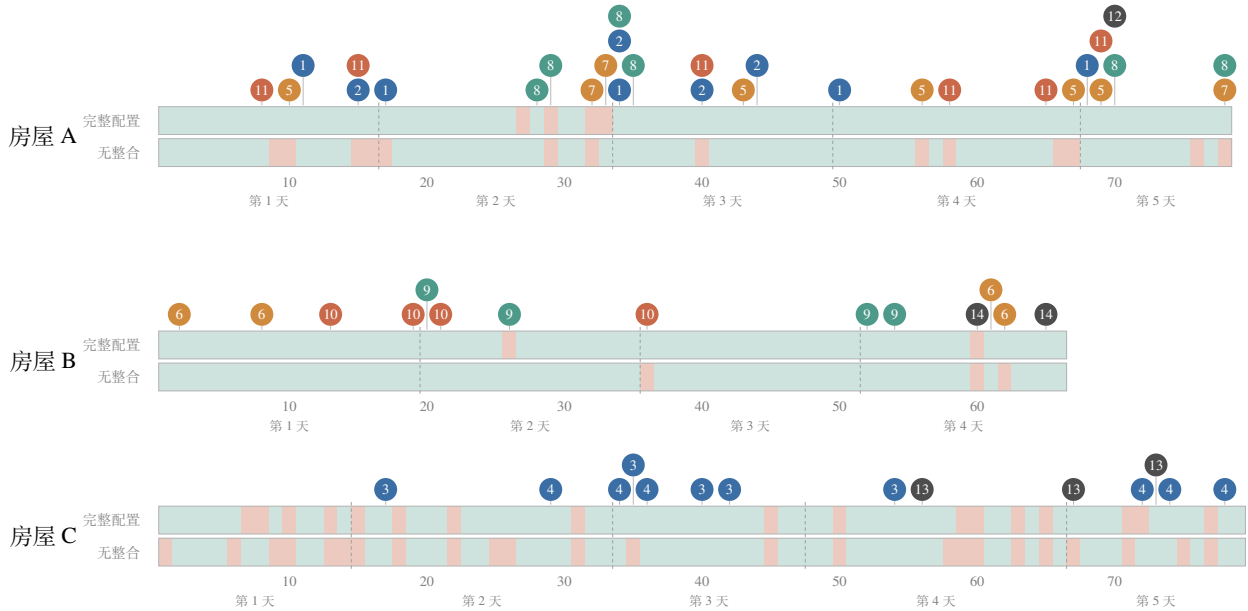


Figure 16 案例在部署 A–C 中的位置。按顺序每格对应一个任务，环境真值下成功为蓝绿色、失败为红棕色；每所房屋的上条带是完整运行，下条带是无整合运行，虚线表示完整运行的日期边界。带编号圆点标出各案例引用的任务：修订空间记录（案例 1–4）、记录失败与检查完成（案例 5–7）、仪器与流程（案例 8–9）、审计与整合记忆（案例 10–11）、拒绝已存经验（案例 12–14）。案例分散在各天，而非集中于每天开头；多数案例对应两条运行作出不同判断的任务。

橱任务可利用机器人脚下的标记作答（案例 6）。房屋 C 的 SPL 曲线多次交叉，增益来自翻转为成功的任务。图 16 将案例标在同一组条带上。

D.2.1 修订空间记录（案例 1-4）

案例 1：随地图坐标系而非物体变化的“修正”。本案例区分运行之间的跟踪器重启与单次运行内的跟踪丢失。保存的标记能在两类事件后保留，但其坐标只有与当前跟踪坐标系对齐时才可用。以下测量追踪完整运行中的重新放置行为。房屋 A 的执行器在部署期间 29 次于已有标签下重新放置标记。每次重新放置可用两种方式测量：地图中新旧标记间的距离，以及模拟器中两次站立位置间的距离。其中 16 次为细化，两种距离都很小；13 次为坐标系事件，其中 9 次发生在每天开始时——SLAM 从原点重启而前一天标记保留原坐标——另有 4 次发生在当天跟踪丢失后；其中 7 次所谓“修正”本身又被放进了已经损坏的坐标系。房屋 B、C 呈现相同模式（图 17）：房屋 B 的 15 次重新放置中，11 次是坐标系事件，2 次是模拟器确认的真实移动；房屋 C 的 34 次中有 21 次是坐标系事件。

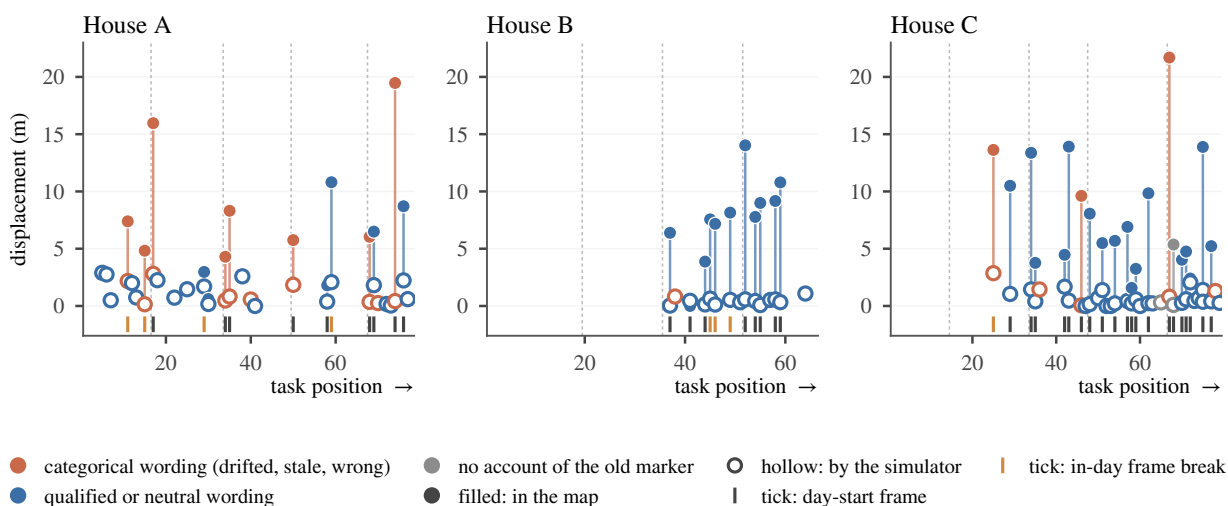


Figure 17 三条完整运行中已有标签的每次重新放置，按任务位置排列。实心点表示地图中的位移，空心点表示模拟器测得的站立位置位移，两点以线连接；颜色表示记录对旧标记的措辞。坐标轴下方刻度标明原因：黑色为每天开始时坐标系重置，橙色为当天内坐标系断裂。类别措辞跟随地图而非物体：地图位移大而真实位移小的事件属于坐标系事件，记录却称其为漂移。

房屋 A · 任务 17 · 第二天 · 完整运行 · 任务记录

当天首个任务：新坐标系中继承标记距地毯 16m

寻找任意地毯。

我在地毯上重新放置了“freezer room”标记——旧标记严重漂移，前往它的路线预览失败。[...] 画面底部被红色填满时，就已经站在地毯上了。

房屋 A · 任务 17 · 无整合 · 任务记录

24 步；在走廊地毯上闭合，被验证器接受，但环境判为失败（4.6m）

寻找任意地毯。

目标是站在走廊地毯上。我已到达——现在站在米色地毯上，走廊两侧是木门 [...] 我在停下处放置了“carpet hallway”标记。尚未探索其他地方。

案例表明。任务记录对观察到的偏移量大小判断较可靠，对偏移原因的判断则不然。在 12 次大偏移中，有 9 次出现 *drifted* 或 *stale* 等类别词；16 次小偏移中有 15 次仅得到保留式评论或完全未被评论。位置记录将大偏移与坐标系变化联系起来，笔记却将其概括为“这里的标记会漂移”。因此，保留标记还需检查其坐标系是否仍然可用。在 13 个坐标系事件任务上，完整运行借助其他保留知识，仍以更少步数完成更多任务。

案例 2：十九个任务后才被发现的错位标记。在房屋 A 中，*laundry bath* 标签首次放置于任务 4，任务 15 又在一个已被跟踪丢失扰乱坐标系中重新放置：地图中新旧标记相距 4.8m，而真实站立位置仅相距 0.15m

(记录称“旧的‘laundry bath’标记漂移到了入口大厅——我在锅炉前重新放置了它”)。直到第三天首个任务 34 才再次使用该标签；每天开始时的重置使继承标记在新坐标系中偏移 4.1 m。执行器沿已存路线前往该处，却落入虚假的“二楼”，请求楼层范围恢复但被拒绝（“机器人创建了‘floor 2’并把我放在上面，而包含我所有标记的整屋地图都在‘floor 1’”）。随后执行器依靠地标走入目标区域并重新放置标签，称旧标签发生了漂移。此后，该标记在当天坐标系中保持准确，路线预览报告其可达，接下来的两个锅炉任务都直接走到该处；任务 44 的记录写道“laundry bath 的标记是准确的”（图 18）。

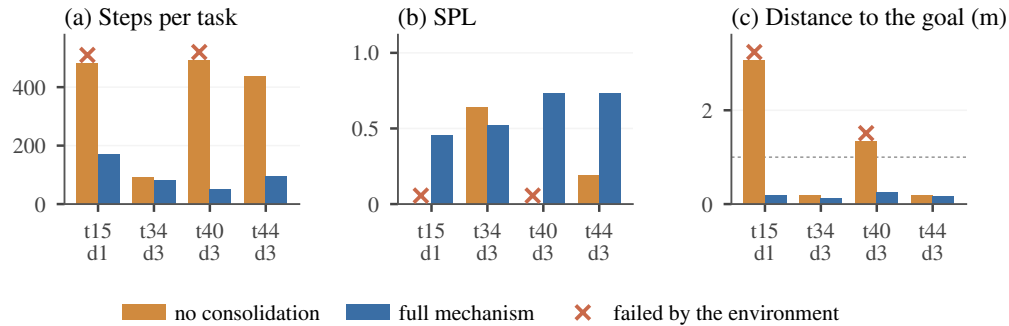


Figure 18 房屋 A 中两条运行放置或使用 *laundry bath* 标记的任务：步数、SPL 与模拟器测得的目标距离（虚线为 1 m 成功半径，叉号表示失败）。任务 34、40 的起点重合；任务 15、44 的起点分别相距 1.4 m 和 0.1 m。无整合运行在任务 15、40、44 各恢复一次，完整运行从未恢复。

房屋 A · 任务 40 · 第三天 · 完整运行 · 任务记录

51 步，SPL 0.73；预览报告标记在 2.5 m 处可达

寻找目标图像中的特定锅炉。

我从一开始就沿已存路线行走——穿过铺砖入口大厅，进入铺地毯的走廊，经过厨房开口和楼梯，再从右侧门进入洗衣/浴室。由于旧标记已经漂移，我在房间内重新放置了“laundry bath”标记。

房屋 A · 任务 40 · 无整合 · 任务记录

491 步，一次恢复，距目标 1.33 m 处失败

相同目标。

我按照照片寻找锅炉，但始终没有找到。我从洗衣房门旁铺地毯的卧室开始，经过有洗衣机和烘干机的洗衣房，沿走廊前进 [...] 检查了沙发房、浴室走廊、红地毯娱乐室，以及摆放卧式冷柜和冰箱的凹室。

案例表明。错位标记首次写入时并未被诊断，在 19 个任务中一直未使用，随后在另一次坐标系重置后才被重新放置（图 20a）。在任务 40、44 中，会话将可达标记与书面地标路线结合使用，且无需恢复：任务 40 分别用 51 与 491 步，任务 44 分别用 93 与 436 步。但任务 34 中，沿已存路线向西反而比无整合多 11 步。这些记录说明，复用曾经成功的路线前检查当前位置十分重要。

案例 3：模拟器从未观察到的“移动”。房屋 C 的冰箱围框上挂着一个日历，相关目标共出现五次。同一标签被放置四次；模拟器显示这些位置彼此相距不超过 0.5 m，地图坐标却随每天的坐标系而改变。任务记录把早期标记称为移动、误导或漂移，三种说法只有一种符合事实；任务 40 中，走 58 步后“已有的‘back hall by fridge’标记就足够了”。图 19 展示两条运行中的全部相关任务；此处讨论的三组任务起点均在 0.9 m 内重合。完整运行在任务 17、35、54、70 重新放置 *back hall by fridge*，地图坐标依次为 (4.54, -3.53)、(1.34, -5.51)、(0.01, 0.02)、(-2.12, -3.41)，并在任务 42 放置 *galley kitchen*。无整合运行每次都重新命名该处 (*kitchen*、*kitchen entry*、*kitchen fridge corner*、*fridge passage*、*fridge end wall*；失败的任务 35 还在全屋放置了四个标签)。无整合运行在任务 40 中第一次从门口闭合被拒后，用 283 步找到日历；任务 54 中则因大批量转向与行走而越过目标。任务 42 中，修复后标记自身的标签反而把完整运行引向错误目标（“把‘back hall by fridge’标记当捷径使我误入歧途”）。

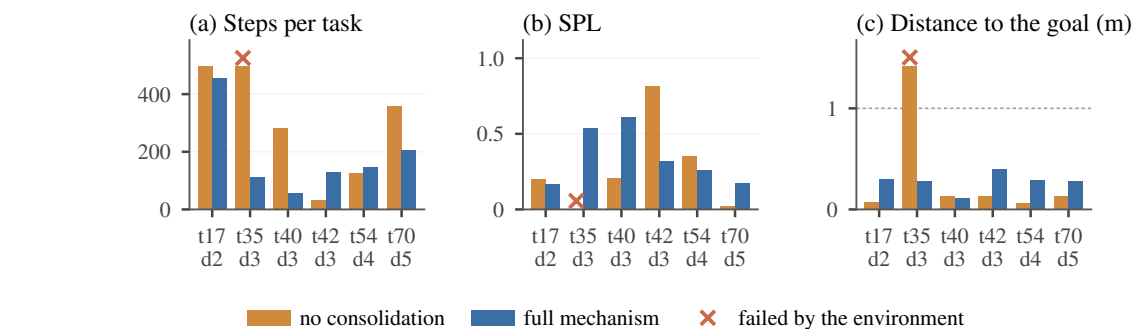


Figure 19 房屋 C 中日历及其旁边冰箱的全部相关任务：两条运行的步数、SPL 与模拟器测得的目标距离（虚线为 1 m 成功半径，叉号表示失败）。

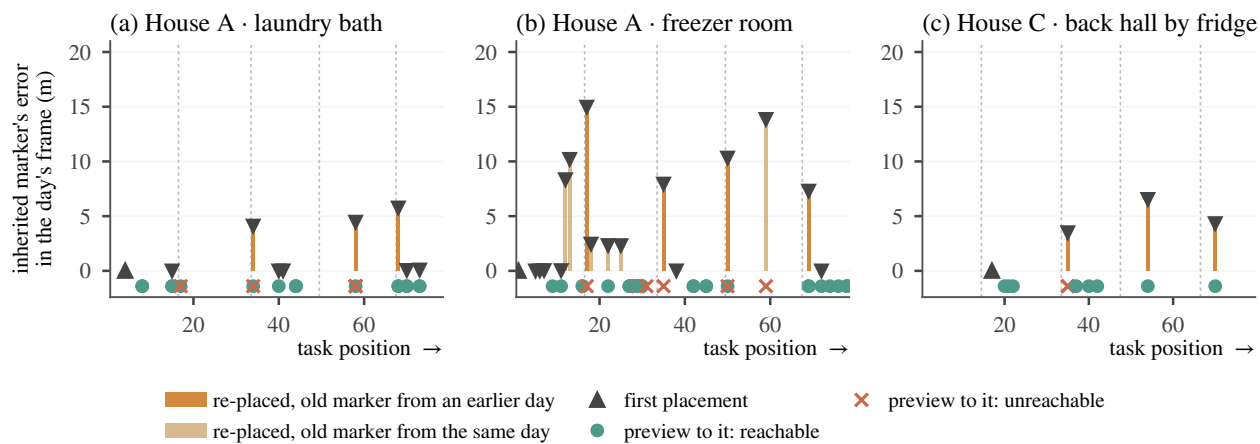


Figure 20 同一标签在每天坐标系中的生命周期。柱条表示继承标记在重新放置时相对于当天坐标系的误差（旧标记来自前一天时为赭色，来自当天时为浅色）；三角形表示放置；坐标轴下方的圆点和叉号表示指向该标签的预览，蓝绿色为可达，红棕色为被拒绝。标记在当天准确，到第二天开始便失准；重新放置前预览始终失败。(a)、(b) 为房屋 A，(c) 为房屋 C。

房屋 C · 任务 35 · 第三天 · 完整运行 · 任务记录

113 步；标记真实移动 0.02 m

寻找任意日历。

我找到了它并站在前面：它位于带饰板的冰箱围框上，在日期方格上方；厨房从这里通向走廊 [...] **路线预览称厨房一侧所有标记都不可达——忽略它。** [...] **已把“back hall by fridge”标记移到能看见日历的位置。**

房屋 C · 任务 35 · 无整合 · 任务记录

496 步，一次恢复，距目标 1.42 m 处失败

寻找任意日历。

我寻找任意日历，但没有找到。从钢琴角开始，我穿过开放式客厅和餐区，进入厨房，再沿走廊前进 [...] 几扇走廊门始终关闭；我在打开或抵达它们之前用尽步数，**走廊墙上的白纸也并非日历。**

案例表明。重新放置可将标记带入当前坐标系，随后它便能指导下一次访问（任务 40 为 58 步）。任务 35 中标记仍不可达，会话转而读取日历技能和四张已存照片。任务 42 展示相反情况：跟随该标签多走了约一百步。这些记录区分了有用的房间识别与不可靠的度量引导。

案例 4：在反复访问中收窄标签位置。房屋 C 的 *piano corner* 标签在第一天放置，之后重新放置六次。每天开始时其地图坐标会跳变 10–13 m；但模拟器显示所有放置点均距目标 2.6 m 以内，最后三次由 1.64 m 收窄至 1.30 m，再到 0.00 m（图 21）。任务 72 放置的位置体现执行器为完整拍下钢琴而后退站立的习惯；该任务在距目标 1.27 m 处失败，而无整合运行用 90 步成功，任务记录却将失败归因于当天记录并不存在的漂移。

两个任务后，执行器在钢琴本体位置重新放置标签并解释原因。任务 74、78 中两条运行均成功，完整运行分别用 136、71 步，无整合运行分别用 171、87 步。

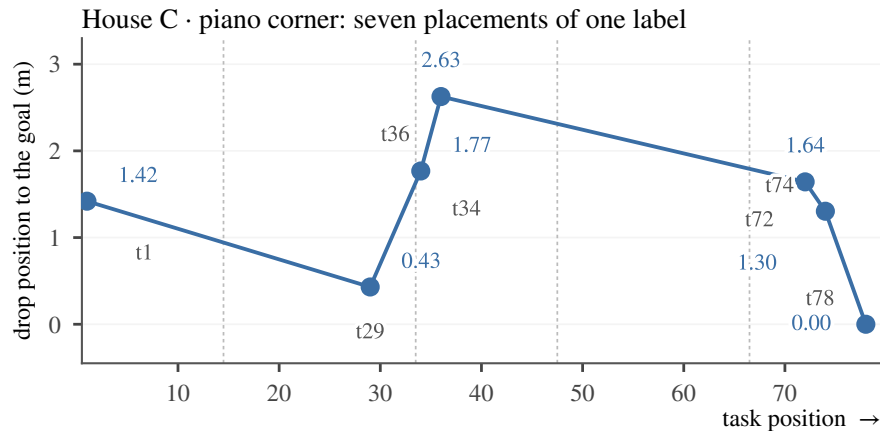


Figure 21 房屋 C 的 *piano corner* 标签：模拟器测得的各放置点至目标距离。相同放置点的地图坐标在不同日期间最多相差 13 m；真实位置均在 2.6 m 内，最后三次逐步收窄至目标。

房屋 C · 任务 72 · 第五天 · 完整运行
寻找钢琴。

到达时的消息及任务记录；运行终点距目标 1.27 m

我在钢琴角，停在约三步之外并正对钢琴 [...] 已在这里重新放置 'piano corner' 标记——旧标记确如笔记所警告的那样发生了漂移。

台账：[...] 重新放置了 “piano corner” 标记。

房屋 C · 任务 78 · 第五天 · 完整运行
寻找目标图像中的特定钢琴。

消息、任务记录及当晚整合的钢琴技能；重新放置点距目标 0.08 m

我正站在目标照片中的钢琴旁。让我把漂移的标记修正到真实位置。

台账：[...] 我在真实位置重新放置了 “piano corner” 标记，因为旧标记已经漂移，把我指向空地板。

skills/find-a-piano.md，第五天：已存的 'piano corner' 标记漂移到了空地板上。

房屋 C · 任务 72 · 无整合 · 任务记录
寻找钢琴。

90 步，距目标 0.08 m，成功

我的目标是钢琴，而且已经到达：我最终站在立式钢琴及其琴凳旁，位置在原木墙客厅的窗边。[...] 我在厨房水槽岛台、原木餐厅和钢琴角放置了标记。

案例表明。重新放置具名标记会更新其存储位置。任务 72、74、78 的最后三次放置依次从 1.64 m、1.30 m 接近至 0.00 m。位置记录表明早期标记代表观察位置而非物体本身，但引文却将其描述为漂移和空地板。完整运行在任务 72 失败，而无整合运行成功；但任务 78 中前者用 71 步，后者用 87 步。七个钢琴任务上，平均步数分别为 74 和 122。标记更新和基于笔记的识别伴随这一改善，但不能保证每次访问都缩短搜索。

D.2.2 记录失败与检查完成（案例 5-7）

案例 5：通过笔记读懂一句描述。房屋 A 中，“木制梳妆台，带镜子，位于狩猎战利品右侧、楼梯扶手上方、架子旁边”这一描述出现三次。无整合运行每次都如实写下排除列表，直到第三次尝试经过 493 步和一次恢复后，才记录“梳妆台一直在地下层，就在我起点旁的楼梯附近”，尽管它此前两个梳妆台任务的成功记录已包含答案。任务 56 中两条运行都从冷冻室内、距目标约一至两米处开始；任务 69 均从锅炉旁开始（图 22）。完整运行除任务 69 的 *freezer room* 外未在这些任务放置标记，其任务 56、67、69 的成功终点彼此相距不超过 0.22 m；无整合运行则在每次失败搜索中留下三个标签（任务 10 的 *front entry foyer*、*office bedroom*、*basement sitting room*；任务 56 的 *red rug freezer room*、*main floor dining*、*upstairs kitchen*）。完整运行通过日志、房间笔

记和技能到达目标：第一天房间笔记写有“带镜子的高木制梳妆台”；任务 43 提示“镜子细节可能已过时”后，第三天技能补充“来自 t10 的修正：从这一侧看不到镜子”；第四天技能则直接把整句描述作为检索键。

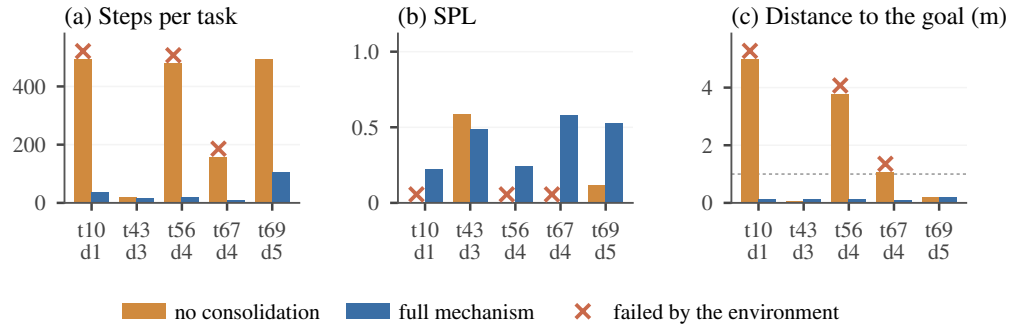


Figure 22 房屋 A 中所有询问带镜梳妆台的任务：两条运行的步数、SPL 与模拟器测得的目标距离（虚线为 1 m 成功半径，叉号表示失败）。

房屋 A · skills/find-dresser.md · 第四天版本及任务 56、69 的开场消息 · 完整运行

寻找这只特定梳妆台：带镜子的木制梳妆台 ...

它的目标文本可能误导：[...] 已认证到这只梳妆台（09-22 t7），尽管从这一侧看不到镜子。应根据上方鹿角饰板、旁边黄色工具架和再往后的角落冷柜来识别——绝不要依赖镜子。

任务 56：[...]（狩猎战利品即鹿角，是冷冻室地标）。

任务 69：我的笔记称，这段完全相同的目标文本已认证到冷冻室梳妆台。

房屋 A · 任务 56 · 无整合

按顺序排列的任务中消息及验证器结果：478 步，一次恢复，在楼上距目标 3.78 m 处结束相同目标。

找到了带扶手的上行楼梯，正在上楼。我在左上方墙上看到像鹿角（狩猎战利品）的东西。我已搜遍整个主层。

验证器：唯一看到的梳妆台状家具（第 7 帧，靠近墙上的战利品）在途中被经过并留在身后。

台账：[...] 唯一的鹿角高挂在入口墙上，下面没有任何东西。

案例表明。无整合运行虽记录了有用信息，却未在后续梳妆台搜索中复用；“楼梯扶手上方”这一短语三次将其引向楼上。完整运行则读取把该措辞与冷冻室梳妆台关联的笔记。任务 43 的观察经任务记录与整合进入技能，之后被后续会话引用。任务 56 中两条运行从同一房间开始，完整会话在移动前直接读取笔记，没有先查询地图。检索到的描述在路线规划前便确定了楼层和目标。

案例 6：一开始就在目标处，却也可能在别处被接受。房屋 B 中，成功率差距很小，但路径效率增益大得多。保留记忆减少了识别并到达熟悉目标所需的移动（图 23）：完整机制下 18% 的任务在十步内闭合，无整合时为 8%。第一天尚无任何笔记，全部差异只来自地图的一行信息——标记坐标等于当前位姿（任务 2：“我醒来时就站在昨天的‘walk-in closet’标记上，面对挂衣杆”，0 步，SPL 1.00；其实该标记是当天任务 1 放置的）。任务 62 是极端案例：房屋有两个挂衣衣橱，无整合运行站在错误衣橱内，走四步后闭合，并凭单帧获得接受；完整运行醒在当天正确衣橱的标记上，无需移动即闭合，尽管它读到的技能笔记指向错误衣橱（“前往‘walk-in closet’标记……预计它可能已过时”）。第四天整合随后写道：“不要先前往 09-19 的‘walk-in closet’标记。”

房屋 B · 任务 62 · 第四天 · 完整运行

按顺序排列的消息及任务记录：0 步，距目标 0.19 m，成功

寻找衣橱中的挂衣。

我正站在先前任务的‘upstairs closet rail’标记上，相机也确认了这一点：外套、衬衫和牛仔裤挂在杆上 [...] 这里就是此前为挂衣目标认证过的位置——无需移动。

台账：值得注意的是，旧技能笔记反而指向灰色卧室的步入式衣橱，在此没有必要。

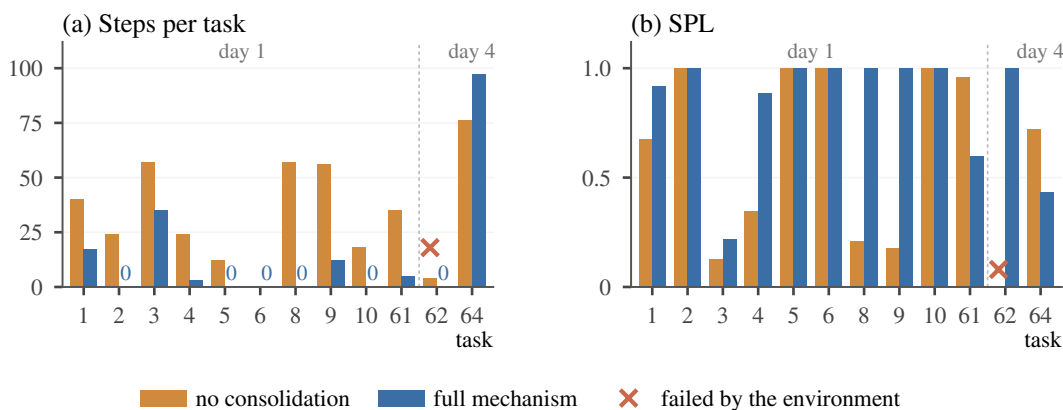


Figure 23 房屋 B 中以文字给出的 12 个挂衣目标任务：先是第一天任务 1-10（目标为图像的任务 7 不计），再是第四天的任务 61、62、64。(a) 每任务步数，零步闭合已标注；叉号表示无整合运行的任务 62：它在错误衣橱中走四步后闭合，被验证器接受，但距目标 9.8 m。(b) SPL。由于任务 61 的轨迹分歧，任务 62 中两条运行从不同衣橱开始，起点相距约 8.8 m。

房屋 B · 任务 62 · 无整合

消息、完成声明、验证器结果及任务记录：4 步，被接受，距目标 9.79 m

相同目标。

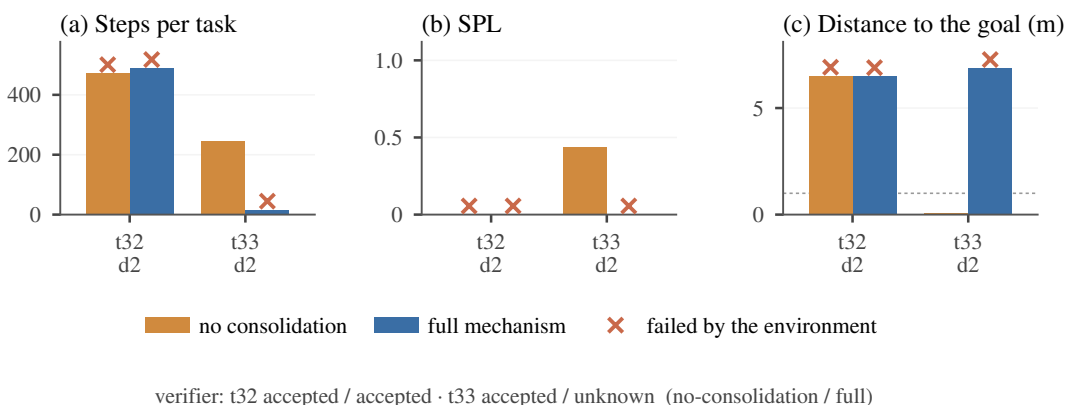
地图几乎空白，没有标记保留下来。重新开始。

验证器：最终帧近距离显示衣橱内衣架上密集悬挂的衣物，与目标相符。

台账：[...] 地图几乎空白且没有保留标记，因此没有先前路线可复用；未尝试其他方法。

案例表明。第一天的任务块中，两条运行起点相距不超过一米。完整会话识别熟悉标记并检查当前视图，避免额外的平移搜索。房屋 B 约三分之一的 SPL 增益来自这一任务块。任务 62 的起点不同，还暴露了另一问题：验证器接受了无整合运行中的错误衣橱。完整会话识别当天标记，将冲突的技能笔记搁置，并把冲突记录下来供整合。因此，接近的总体成功率可能掩盖显著的搜索效率差异。

案例 7：错误完成被认证后写入记忆。本案例展示错误物体识别如何通过闭合检查并进入记忆。房屋 A 中，检查器在无整合时错误接受了 10% 的任务完成，启用整合时为 3%。下方无整合轨迹中，第一次请求被拒后又重复声明，而停止规则允许这种行为（附录 A.6）。启用记忆时，一次错误完成以 CERTIFIED 状态进入日志，被写入客厅笔记（“CERTIFIED, 09-20 t16：一只上放台灯的木制抽屉柜”）和梳妆台技能（“dresser next to the armchair”的答案）。下一任务未读取文件便零代价复述该结论；第五天，它又使任务 78 偏离 26 步（“这与我客厅 ‘dresser next to the armchair’ 的笔记吻合。让我靠近”）。



verifier: t32 accepted / accepted · t33 accepted / unknown (no-consolidation / full)

Figure 24 房屋 A 中扶手椅旁的灯桌。两条运行均在同一物体上闭合任务 32，任务 33 从任务 32 的终点开始；各面板下方给出无整合运行和完整运行每次闭合的验证器判定。

房屋 A · 任务 32、33 · 第二天 · 完整运行

完成声明、台账索引条目、下一任务首条消息及验证器结果

寻找这只特定梳妆台：扶手椅旁的梳妆台。/寻找任意梳妆台。

我站在一只上放台灯的木制抽屉柜旁，它紧邻黑色皮质躺椅。

日志：任务 16 · 已认证 · [...] 标记：客厅。

任务 33：[...] 这里就是任务 16 的客厅角落。

验证器：它看起来更像床头柜或边桌，而不是梳妆台。

房屋 A · 任务 32 · 无整合

停止前拒绝、第二次声明获接受及任务记录；距目标 6.51 m，失败

寻找这只特定梳妆台：扶手椅旁的梳妆台。

门控：未看到梳妆台，也未看到与其相邻的扶手椅 [...]

台账：我在客厅的黑色皮质躺椅处将任务闭合为完成；旁边带灯的柜子位于画框下方——但检查器曾表示不同意，而我覆盖了其判断，因此该识别并不可靠。

案例表明。错误认证使一项物体识别作为事实保存在房屋笔记中。下一任务的“unknown”判定记录了该任务的不确定性，却不会自动撤回先前记录。

D.2.3 仪器与流程（案例 8–9）

案例 8：同一故障诊断一次，还是十次。房屋 A 中，跟踪丢失后跟踪器会将位姿跳回原点。两条运行都认识到地图正确而位置错误。无整合运行在十个任务中反复重新诊断，并给出相互矛盾的补救措施；完整运行则保留了“预览不可靠时跟随地标”的建议。更精确的解释出现在被拒绝的恢复请求、Edit 和 Write 中，已保留技能仍只写着这里的标记会漂移。任务 35 的一次被拒编辑指出“相对于旧的一楼地图存在坐标系偏移”；任务 78 中，运行仅凭视觉熬过了位姿冻结：“位姿读数冻结，但视图确实在变化——我会相信视图”。

房屋 A · 任务 29、34、70 · 完整运行

三次正确诊断、三次请求被拒；最后引用保留下来的笔记

任务 29：[...] 地图把两个半层合并到了一起。

任务 34，恢复请求被拒：机器人创建了‘floor 2’并把我放在上面 [...]

任务 70，被拒绝的 Write：标记没有漂移。[...] 规则：标记只能用于确认已经到达，绝不能用于选择朝向。

skills/verify-markers-before-trusting.md，第五天：即使标记昨天一整天都准确，也可能失效。

房屋 A · 任务 28 · 第二天 · 完整运行

消息、任务记录及由此形成的规则：规则链的第一环

寻找任意锅炉。

当我走进先前未建图区域时，位姿突然向西跳变。

台账：进入途中 SLAM 丢失定位，标记位置看起来不对；恢复没有改变任何情况，因此应相信路线而不是标记。

skills/ignore-route-preview.md，第二天：如果 SLAM 看起来丢失，应继续沿地标链行走，而不是尝试恢复——t12 中恢复毫无作用。

房屋 A · 任务 71 · 无整合 · 任务记录

429 步，204 轮，SPL 0.01；补救方法与任务 49 矛盾

寻找目标图像中的特定锅炉。

我始终无法确认已到达——移动不断停滞，前进动作持续碰撞，位置反复跳回起点，因此洗衣凹室的同一视图不断重复出现。[...] 原地转向寻找通路基本没有帮助。

案例表明。两条运行都能诊断反复出现的跟踪故障。完整运行通过源自任务 28 的预览技能复用早期诊断，整个部署仅记录 7 次恢复请求，而无整合运行行为 13 次。任务 78 中，即使位姿冻结，完整运行仍利用变化的相机视图继续；无整合运行在任务 71 遭遇类似故障时请求恢复，用去 429 步且 SPL 仅为 0.01。这些来自不同任务的例子并非匹配任务效应估计，但说明已存流程如何改变系统对反复出现的仪器问题的响应。

案例 9：虚假楼层得到诊断，却从摘要中丢失。只有带记忆的运行会遇到此故障：标记跨天保留，但 SLAM 每天开启新坐标系，楼层计数器也随之递增，于是全部旧标记落入失效坐标系，预览报告其不可达。房屋 B

的执行器在第二天故障出现时立即将其写成错误报告；到第四天，它已掌握机制（任务 52：“会话间坐标系发生偏移……它指向失效坐标系中的位置”）、补救方法以及“不要为此请求恢复”。详细诊断出现在被拒绝的编辑尝试中，而下方保存的任务记录只写了标记不可达和改用地标。之后的技能笔记保留了补救方法，却遗漏坐标系变化的解释。整合可以读取台账、任务交接记录和既有笔记（附录 A.4）；此处的问题是保留摘要丢失解释，而非系统只能读取台账。

房屋 B · 任务 20 · 第二天 · 完整运行

故障发生时的恢复请求、被拒日志草稿及最终进入日志的任务记录

寻找任意微波炉。

仅原地转向后，`get_map` 就从 'floor 1/1' (127 m²，包含我所有标记) 切换到全新的 'floor 2/2' (17.85 m²，无标记)，前往厨房标记的 `preview_path` 也从可达 (12.5 m) 变为不可达。

被拒绝的日志草稿：地图现在显示两个楼层图层 [...]

台账：旧厨房标记不可达，因此我改用地标导航。

房屋 B · 任务 54 · 第四天 · 完整运行

被拒绝的 Edit，以及第五天整合后技能笔记的首行

会发生整体坐标系偏移，而不只是单个标记失效。[...] 因此移动的是地图原点，而不是家具。[...] 不要为此调用 `request_recovery`——机器人的真实位置没有问题。

第五天笔记：已存标记位置会失效。[...] 忽略预览，沿地标走入目标区域，并在终点用同一名称重新放置标记。

案例表明。执行器三次诊断坐标系变化，解释越来越具体，但保留笔记只保存补救方法而未保存原因。记录的图层变化并不对应真实楼层。因此，原本针对坐标系重置的警告被泛化为对路线预览的普遍不信任，即便当前坐标系可用也如此。轨迹中包括任务 20 的额外步数与一次恢复、任务 26 的失败，以及第四天每任务 4–8 次预览探测。摘要不仅要保留补救方法，还应保留其适用条件。

D.2.4 审计与整合记忆（案例 10–11）

案例 10：关闭一个待解问题。问题并非由执行器提出；房屋 B 第一天的整合并列放置了厨房冰箱的两种描述，询问它们是同一电器的两个侧面还是两台电器（“仍待确认：厨房有一台还是两台冰箱”；`house.md`：“可能是从两侧看到的同一电器，也可能是两台”）。第二天，执行器读取该问题，完成任务后又三次读取概览，并查看电器的两个侧面。答案经过台账、当日摘要和技能传递，没有丢失。两天后出现一张很暗的目标照片；无整合运行根据黑暗环境推断目标在地下室，转而寻找第二台冰箱，最后停在目标旁边。完整运行在任务 21 放置 *kitchen dining doorway*，任务 36 放置 *fridge nook*；无整合运行每次成功都放置 *kitchen* 或 *kitchen fridge*，失败的任务 36 则放了四个标签，最后一个是 *basement stairhead*。任务 21 中，完整运行行为回答待解问题用去 56 步中的 36 步；任务 36 中两条运行从同一位置开始（图 25），完整运行的台账写道：“旧笔记说得对，应该从餐厅门口一侧靠近，而不是从炉灶正面接近”（36 步，SPL 0.70）。

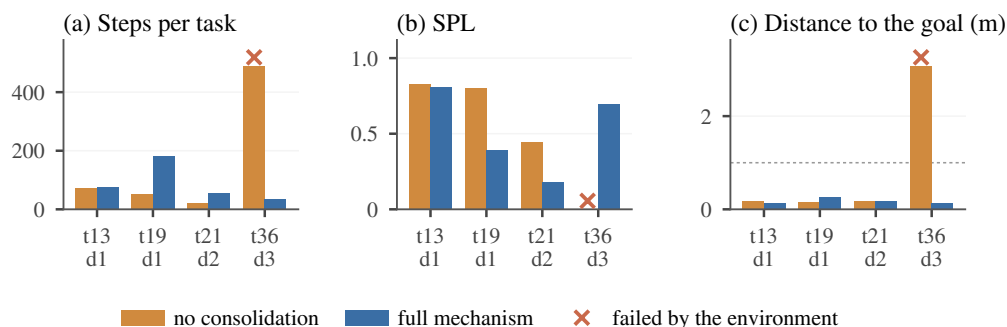


Figure 25 房屋 B 中所有询问厨房冰箱的任务：两条运行的步数、SPL 与模拟器测得的目标距离（虚线为 1 m 成功半径，叉号表示失败）。任务 36 的目标是一张暗色照片。

房屋 B · 任务 21 · 第二天 · 完整运行

任务完成后的消息、被拒绝的写入及任务记录；house.md 又被读取三次

寻找这台特定冰箱：...

我要解决笔记中长期悬而未决的问题，然后闭合任务。[...]

被拒绝的写入：“门边的深色高冰箱”（任务 13）与“带冰水分配器的不锈钢双开门冰箱”（任务 19）是同一台电器，而不是两台。[...]

台账：深色侧板与不锈钢正面属于同一台冰箱，而非两台。

房屋 B · 任务 36 · 无整合

消息及任务记录；490 步，一次恢复，距目标 3.08m，失败

相同目标。

目标图像非常暗 [...] 这是一台位于无照明空间中的深色冰箱——几乎肯定在地下室，而不是这个明亮厨房。

台账：我没有找到它。[...] 相连厨房中的冰箱亮白，显然不是图片里的那台 [...] 深色冰箱最可能位于地下室楼梯下方。

案例表明。整合将冲突描述转化为待解问题。执行器收集额外视图解决问题，答案在后续摘要中得以保留。任务 36 中，完整运行把暗色目标照片识别为已知冰箱的另一视角，无整合运行却去寻找第二台电器；两条运行分别用 36 步和 490 步。轨迹将已存答案与目标解释联系起来，但不会把 454 步差异进一步分摊给笔记、地图和其他保留状态。

案例 11：被推翻的规则与无人读取的路线。房屋 A 的七条技能笔记都有版本历史，每次推翻都可追溯证据来源。由执行器自身观察推动的三次修订（地毯停止规则、镜子可见性、裸露顶盖判据）是正确的；由验证器判定推动的一次修订是错误的——某个已被模拟器认证三次的目标出现一次“unknown”后，第五天整合便反转规则，如今该措辞会把执行器引向另一楼层。由执行框架自身伪象（“死胡同”“漂移”）推动的两次修订，则以错误理由保留了正确做法。无整合记录包含后续会话未在这些搜索中复用的有用经验：任务 8 完整写下的锅炉路线从未被读取，之后三个锅炉任务都把全部预算耗在新假设上（任务 15：“工作台拐角后面的储物空间是我唯一没进过的杂物间”；任务 58：“天花板管道朝那边延伸，所以锅炉可能在后面”）。九个锅炉任务上，无整合运行成功率为 89%，平均每任务 274 步；完整运行九个全部成功，平均 107 步。

房屋 A · skills/find-dresser.md 第五天版本及推动修订的任务 69 验证器判定 · 完整运行

模拟器此前已三次确认冷冻

室梳妆台

完全相同的措辞曾获一次认证（09-22 t7）；该物体上看不到镜子，09-23 t2 的重复尝试也未解决。如果文本同时提到镜子和战利品，优先选择靠窗卧室的柜子，并注明冲突。

验证器，任务 69，未知：[...] 但没有看到镜子或可辨认的完整梳妆台。

房屋 A · 任务 65 · 第四天 · 完整运行

读取到的裸盖规则、消息、任务记录及第四天整合

寻找目标图像中的特定冷柜。

技能，第三天：[...] 已认证答案是右侧冷柜，而不是角落那台。

近距离顶盖 [...] 我的笔记说是右侧冷柜。[...] 这与黄色架子下方的角落冷柜吻合。

台账：我那条“裸露顶盖意味着右侧冷柜”的规则差点误导我；角落位置与锈斑才是决定性证据。

第四天整合：[...] 对该规则保持怀疑，应根据锈斑与角落位置判断。

房屋 A · 任务 8 · 无整合 · 任务记录

一条未在后续锅炉搜索中复用的完整路线

寻找这台特定锅炉：燃气锅炉 ...位于加热器下方略偏右 ...

已找到并站在旁边——这是带燃气阀的直立圆筒，位于炉管下方、洗衣机和烘干机旁。路线从起点冷柜凹室出发，沿走廊经过一个门口，再右转进入尽头的洗衣杂物间。

案例表明。来源引用使修订可审计。在这些例子中，直接观察纠正了误导性规则，不确定的验证器输出却可能使整合用错误规则替换有用规则。无整合运行也写下完整锅炉路线，但后续锅炉搜索并未复用。完整运行在锅炉任务中把记忆中的地标路线与当前标记结合，平均用 107 步而非 274 步。这是系统级对比，不能将测得的步数节省归因于某一句未读内容。

D.3 拒绝已存经验时（案例 12–14）

以下案例展示三种选择性复用形式，均有无整合对照：依据笔记中的先验拒绝一条路线（案例 12）；因目标与规则矛盾而拒绝规则（案例 13）；根据不良条目的后果发现问题，再利用日志修复（案例 14）。对照组保留地图和任务记录，但没有整合后的房屋笔记；其引文中的疑虑仅涉及任务内位姿与预览。

案例 12：拒绝正确标记，改走记忆中的路线。房屋 A 的任务 70 中，标记误差仅 0.06 m，预览也报告可达；但笔记先验称该标记“以前漂移过”，执行器因此拒绝直线前往，转而沿地标链行走，并在到达后纠正原因（被拒绝的写入：“我的旧笔记 #8 声称 ‘laundry bath’ 标记发生了漂移。其实没有”）。台账保留了这一撤回（“标记是准确的，但只有走廊路线能到那里”）。两条运行都从冷冻室梳妆台旁开始，最终均到达锅炉 0.2 m 内；完整运行用 94 步（SPL 0.80）且未恢复，无整合运行用 317 步、恢复一次，SPL 为 0.20。

房屋 A · 任务 70 · 第五天 · 完整运行

读取笔记且预览报告标记在 4.7 m 处可达后，按顺序排列的任务中消息

寻找任意锅炉。

距离 4.7 m，可达——**但我的笔记警告这个标记以前漂移过**。[...] 标记正指向梳妆台所在墙面——它确实漂移了，与笔记警告一致。改用地标路线。[...] 黄色板条松木门在左侧——这是关键地标。[...] 到达时：**标记其实一直准确（相距 0.36 m）**——我的直线方位穿过了墙壁。

房屋 A · 任务 70 · 无整合

跟踪丢失后的任务中消息及任务记录；317 步，一次恢复，SPL 0.20

寻找任意锅炉。

报告的位置显然错误——地图称我在娱乐室原点，但我实际位于西侧格纹沙发房。

台账：[...] **早期我在格纹沙发房附近的一面墙旁迷失方向，浪费了若干转向动作重新定向，之后走廊路线才奏效。**

案例表明。由于旧笔记警告存在漂移，且目标直线方位穿墙，会话起初不信任正确标记；它沿记忆中的地标路线行走，并在到达后修订诊断。两条运行均成功，分别用 94 步与 317 步。轨迹说明，有用路线可能与选择该路线的错误解释并存；它并不能证明，相较于结合有效路线使用标记，拒绝标记毫无代价。

案例 13：拒绝与目标照片矛盾的规则。整合可能把“我不需要这样做”变成“不要这样做”。房屋 C 的任务 56 记录称，角落的两盆落地植物太挤，无法完整取景，因此选择了另一盆；第四天技能把这条经验写成避免。任务 67 的目标照片恰是其中一盆。两条运行都从该回合的重置点开始；完整运行用 123 步成功（SPL 0.75，距植物 0.17 m），无整合运行失败。

房屋 C · 任务 56 · 第四天 · 完整运行 · 任务记录

后来被整合转化为禁止性规则的记录

寻找任意装饰植物。

我从东北角的脚踏桌凹室开始；那里有两盆落地植物——木桶里的干花和观景窗旁的低矮绿叶盆栽——**它们在角落里太挤，无法完整取景，所以我离开了**。随后我穿过 [...] 前往餐厅窗边凹室。

房屋 C · skills/find-a-decorative-plant.md 第四天版本、任务 67、73 及第五天版本 · 完整运行

寻找目标图像中的特定装饰植物。

避免：两盆落地植物 [...] 位于过于狭窄的角落，无法完整取景——09-22 t9 放弃了它们 [...]

任务 67：**旧笔记说该角落太挤而无法取景，这是错误的**；站在离桌子一步处即可完整取景。

任务 73：[...] **我无法保存这项修正；笔记写入工具已禁用，因此过时建议仍在文件中。**

第五天：旧警告 [...] 已失效——在离桌子约一步处停下即可取景（09-23 t7，t9/t13 也证明如此）。

房屋 C · 任务 67 · 无整合 · 任务记录

470 步，一次恢复；距目标 7.1 m 处闭合，判定未知

相同目标。

匹配的植物放在一只木柜上，旁边是黄色灯；位置在餐厅之后、采光明亮的窗边客厅。我在那里闭合了任务。 [...] 在尝试最南侧房间之前，我在厨房和壁炉区域兜圈，浪费了许多步。

案例表明。会话利用已存房间描述与照片识别该角落，但由于新目标明确要求寻找那盆植物，拒绝了避开该处的指令。它在任务记录中写下冲突，整合随后修订规则。关键行为是选择性复用：即使相关指令不适用于当前目标，房间信息仍然有用。

案例 14：笔记造成误导，日志帮助恢复。第四天，房屋 B 的笔记只提到一面镜子，即入口大厅中的黑框镜子。系统在任务开始、恢复前和恢复后都返回了目标图像，但执行器仍按笔记下楼，随后还写道本会话没有见过目标照片。直到第四次调用，它才查看目标（“重新读取真实目标照片，而不是相信我的笔记”），发现目标并非那面镜子；随后又在当天日志中找到五个任务前同一目标的失败尝试，其中留下了精确线索。记忆既是问题起因，也是补救手段。两条运行起点相距 1.2 m，均位于同一客房。

房屋 B · 任务 60 · 第四天 · 完整运行 · 写入日志时的任务记录 500 步，一次恢复，距目标 1.21 m，失败
寻找目标图像中的特定镜子。

我按照照片寻找木框镜子，但没有找到。我醒来时位于楼上卧室层，而不是旧笔记所述的主层 [...] 穿过主卧门后，硬木走廊经过一段下行楼梯，通向一间放着蓝绿色床的客房；我已标记该房间。步数耗尽前，我没能看到敞开房门后的墙面。下次先检查那面墙。

房屋 B · 任务 65 · 第四天 · 完整运行 第 241 步的恢复请求及后续消息
当天五个任务后的相同目标。

本会话中我仍未真正查看目标照片——一直在依据笔记推理。[...] 真实目标图像是一面松木框方镜，挂在石板蓝灰色墙上 [...] 并非入口大厅那面镜子（黑框、奶油色墙）。
读取日志：任务 9 就是同一镜子目标，而且留下了精确线索：“[...] 我在看到敞开房门后的墙面之前用尽了步数。” [...] 我需要回到楼上。

房屋 B · 任务 65 · 无整合 · 任务记录 156 步，SPL 0.38；从未下楼
相同目标。

我在有红色书架的卧室里找到了它：它挂在蓝灰色墙上，旁边是一幅小装框画，正对通往走廊的敞开房门——我面向它停下并闭合为完成。[...] 在第一个房间徘徊白白浪费了步数。

案例表明。尽管已经收到目标图像，会话仍把不完整清单当成穷尽列表，追随笔记中的唯一镜子。恢复后，它重新检查图像，并从日志检索任务 60 的未完成搜索线索；该线索引导搜索在 468 步后成功。不过，完整运行仍低效于无整合运行（SPL 0.08 对 0.38）。详细失败记录能修复被摘要误导的搜索，但不能说明缺少这些记录时另一次失败必然发生。

发生多频繁，判断是否正确。本文统计完整运行的记录、消息、声明、恢复请求和被拒绝写入中，所有把继承条目判为错误、不可靠或不适用的陈述，并用模拟器核验（图 26）。该分析关注完整运行实际记录的判断，而不是对照组能否使用记忆。三个房屋中，“结果正确但原因错误”指所述偏移或不可用确实存在，却被错误归因于漂移或楼层不对。房屋 A 对继承标记有 17 次怀疑：9 次正确、4 次效果正确但原因错误、4 次错误；对笔记中规则、路线或目标有 8 次怀疑：3 次正确、3 次错误、2 次无法核验。房屋 B 的 21 次可核验判断在效果上全部成立，其中 2 次原因错误；房屋 C 的 17 次可核验判断（共 19 次）中，8 次成立、7 次成立但原因错误、2 次被事实否定（任务 36、72）。

拒绝继承坐标在效果上几乎总是正确，但原因判断几乎从不正确：每天开始时的坐标系重置都被记录为漂移或楼层错误。三次错误判断怀疑了准确标记（房屋 A 的任务 3、40、70），另一次导致任务失败（房屋 A 的任务 27）。统计只包含书面判断；未被使用但也未说明的记忆不会留下记录，模拟器也只核验位置。

所分析案例中的记忆使用。对于 48 个案例，本文人工将记录中的记忆效应编码为正向、负向、混合或无差异。同时标注所涉及的记录；笔记、当天标记和台账可多选，执行框架或判别器参与另设标签。图 27 汇总这些案例级解释。它们描述的是已分析语料，而非独立消融效应或全部任务上的发生率。

对比表明。四类笔记内容会指导记录中的决策：将目标句映射到物体的规则（案例 5、10）、房间清单与照片（案例 3、13）、技能中的地标链（案例 11、12），以及被记录为已认证、因而压过错误技能的地点（案例

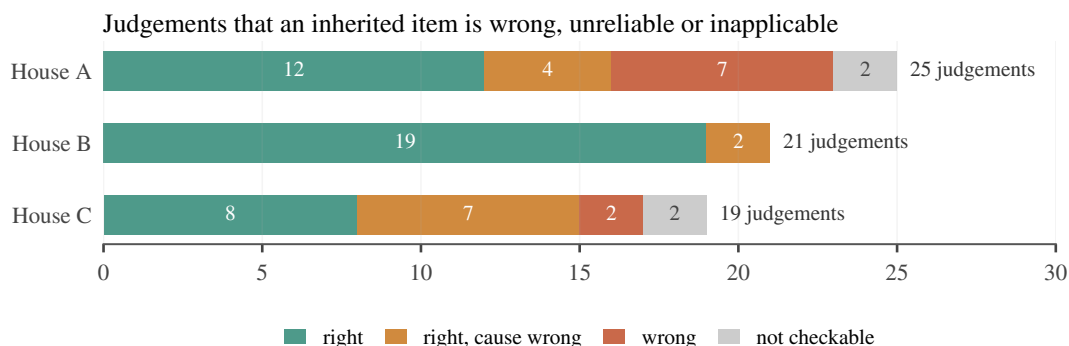


Figure 26 完整运行中将继承条目判为错误、不可靠或不适用的判断，并用模拟器核验。计数描述完整运行实际记录的判断。

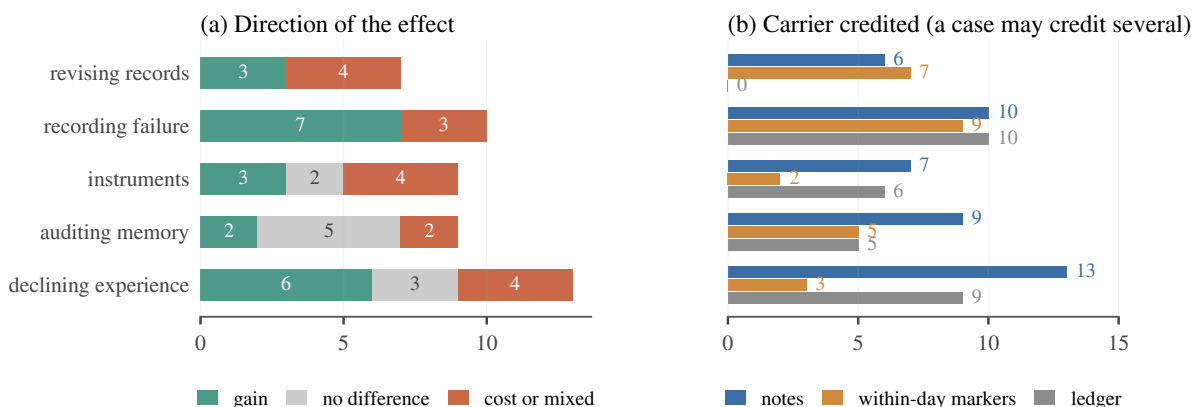


Figure 27 对 48 个分析案例（包括本文展示的 14 个）的人工编码。(a) 记录中记忆效应的方向；(b) 案例评估涉及的记忆来源，可多选。计数描述案例，而非任务级组件贡献。

6)。这些文本线索与地图及机器人当前位置协同使用。当天标记承载另一类信息：醒来时就在目标上，或标记在当天坐标系中可达（案例 2、4、6）。台账把当天观察转化为第二天笔记（案例 5、10、13），也可能让错误观察进入笔记（案例 7、11）。记录还揭示复用记忆的五类风险：每天开始重置坐标系后继承旧坐标（案例 1、2、3）、虚假楼层（案例 8、9）、被接受的错误完成以认证事实写入（案例 7）、笔记缺口或失败记录中的线索把搜索引入死胡同（案例 14），以及把“不需要”泛化为“禁止”（案例 13）。这些行为有助于解释图 12 中的不同画像，但不把任何画像归因于单一记忆组件。

D.4 档案构建与访问

部署期间，长期记忆通过房屋概览、房间笔记和技能文件，将任务经验跨模拟日保留下来。概览是进入这些内容的入口，而非详细记录的替代品；执行器通过文件工具读取记录。任务记录按产生顺序保存新经验，独立的整合会话则将经验概括后供后续使用。第 4.4 节的 36 所房屋对比固定地图与任务记录的保留方式，仅关闭整合会话。启用整合后，汇总 s-SR 从 72.8% 提升至 80.5%，配对增益为 7.7 个百分点（95% CI [+6.3, +9.1]）；SPL 从 36.5 提升至 44.3，增益为 7.8 个百分点（95% CI [+6.5, +9.2]）。因此，即使地图和详细任务记录仍可用，房屋笔记也能改善后续导航。附录 D.2 的案例沿着“观察—后续决策”追踪这一过程，包括修正梳妆台描述、通过多视图确认冰箱身份，以及修订植物位置说明。档案同时保存成功路线、失败搜索、待解问题及其来源，使后续会话可以重新核查笔记背后的证据。

E 局限性与未来方向

部署结果指出了扩展 NavHarness 的若干方向，包括更快速的局部控制、记忆访问以及对变化环境的适应。

部署证据的范围。连续部署比较涵盖多个静态仿真房屋中的有限任务序列。实验固定地图与任务记录保留、恢复和完成检查，仅考察运行结束时的整合。基准干预则分别测试地图和台账复用。先前轨迹一旦分化，配对目标序列未必产生相同的起始位姿。因此，路线分析还考察两次运行均从相近起点完成的任务。案例研究旨在解释记录下来的决策与记忆修订，而非估计每种机制在更广泛住宅总体中的出现频率。将研究扩展到变化环境和更长期的部署仍然很重要。

E.1 控制循环的归属

导航由反复的模型决策推进。在部署 A–C 中，完整系统每项任务平均使用 38–45 个模型轮次。一个轮次可发出包含多个原子动作的工具调用，因此模型轮次、工具调用和移动步数是不同的量。更强、更快的具身策略可以接管局部展开，由更大的模型维护目标、记忆和任务边界。这要求局部控制器既能可靠导航，又能返回支持后续推理与验证的观测。将执行器与具身执行框架分离，便可在不改变持久状态及其来源记录方式的情况下替换执行器。

E.2 执行速度与物理时间

即使批量执行动作，反复的模型推理与工具交互仍会引入显著延迟。任务边界、恢复和记忆整合均由事件触发，因此其调度也可纳入经过时间，而无须改变存档格式。

E.3 真实世界评测范围

本文报告的评测均在仿真中进行。基于传感器的 SLAM 消除了对仿真器位姿的依赖，但物理部署还要求稳健通信、低延迟避障以及独立测量的轨迹。收集并核查一整天的真实世界观测与结果，仍构成沉重的评测负担。

E.4 空间表示

当前空间状态使用 ORB-SLAM3 地图、命名标记以及小规模地点照片集。它无法表示物体运动、时间变化或可操作物体的状态；其度量坐标系也从未重新锚定到真实值，因而漂移与坐标错位可能持续存在。此外，跟踪器的回环闭合校正不会传播至先前写入的占用栅格。识别熟悉地点与恢复其度量位姿是两项不同要求。地点识别支持复用房间层级知识，而基于坐标的导航还要求与存储地图可靠对齐。更丰富的后端可提供稠密 3D 几何，或像 SuperMap (Zhao et al., 2026) 那样提供感知变化的 4D 记忆，以跟踪物体的出现、消失与重定位。集成这些表示需要调整地图查询与更新工具。具身执行框架规定保留哪些状态、何时更新以及如何记录其来源，而不要求特定的地图表示。

E.5 长期部署中的存档访问

存档贯穿多轮巡视部署运行，更新后的房屋概览会进入后续会话。整合过程将任务经验组织为带来源引用的房屋、房间与技能笔记，这些概览构成导航器后续输入的一部分。整合对比表明，其收益超出了保留地图和任务记录本身。部署时间越长，访问策略越重要，包括何时检索详细记录、何时修订旧有认知。当前设计向会话提供房屋概览，并支持按需回忆单条笔记。随着经验积累，检索必须保持足够的选择性，使相关信息仍可访问，同时避免活动会话过载。部署案例还表明，处理过时或相互冲突的记录同样重要。

E.6 静态环境与部署时长

评测采用 GOAT-Bench 和 IR2R-CE 的静态扫描环境。扩展部署测试了有限任务序列和预设重定位之间的记忆复用，但未评估对物体移动、布局变化或人员的适应能力。尽管智能体会遇到不完整或错误的记录，修订这些记录并不能证明系统能够检测并跟踪环境本身的变化。未来仍需在环境变化与更长期部署下评估记忆修订能力。

E.7 较小模型的能力

固定使用 Qwen3.8-27B 时，具身执行框架将 GOAT 任务成功率从 41.4% 提高到 71.7%。这表明其价值不局限于最大的前沿模型，但仍要求执行器具备多轮多模态导航能力。在测试中，Qwen3.5-4B 和 Qwen3.5-9B 难以完成多轮多模态导航，任务完成率较低。将这些模型用作共同骨干，既会测试推理核心的局限，也会测试具

身执行框架。因此，我们在四种能力足够的执行器中分别比较具身执行框架与独立会话配置。使接口适配较小模型仍是未来工作。

面向未来训练的过程记录。这些运行将目标、帧、执行记录、声明、检查判定和记忆更新对齐。对并不完全可靠的判定器标签进行独立验证后，这些记录可支持长时程后训练或过程监督。当前研究不更新模型参数。

E.8 监督位置

基准评测与连续部署都会检查首次停止请求，并评估已完成的展开。其他设计可以每隔固定步数检查一次，或持续监控执行器。频繁检查可能中断有效工作并增加判定器调用次数；持续监督则需要全双工接口和低延迟验证器，而当前传输机制均不支持。这一选择也与循环归属相关：由局部策略负责展开时，控制权返回的边界是自然的检查位置，三种调度因而成为同一停止前验证机制的三种触发规则。验证调度可独立于执行器和记忆保留规则进行更改。

E.9 仿真与基于模型的判定

基准场景标识符用于选择房屋存档，而真实机器人则需要跨房屋地点识别。最后，学习得到的判定器可能产生幻觉，并与执行策略共享模型家族偏差。尽管其声明完成准确率高于全接受策略，四视图检查器仍会接受 5.2% 的错误声明，并拒绝声明完成中 4.1% 的真实声明（附录 C.5）。目标验证也未验证记忆中的空间与过程性声明。本文报告的导航分数来自环境真实值。我们将失败或无法解析的判定映射为 `UNRESOLVED`，且不把连续部署结果视为基准排行榜分数。真实机器人运行、地点识别以及独立训练的判定器仍是必要测试。