arXiv:2603.04370v1 [cs.AI] 4 Mar 2026

τ-知识：评估 非结构化知识上的对话智能体评估 e

Quan Shi *[1] **Alexandra Zytek** *[1] **Pedram Razavi** [1] **Karthik Narasimhan** [2] **Victor Barres** [1]

§ 代码

摘要

对话智能体越来越多地被部署在知识密集型场景中，其正确行为取决于能否在与用户实时交互时，从大型、专有且非结构化的语料库中检索并应用领域特定知识。然而，现有的大多数基准将检索与工具使用分开评估，因而在长程交互中针对非结构化数据进行逼真、完整的智能体评估方面存在空白。我们推出 τ-**知识**，这是 τ-基准的扩展，用于评估这样一类环境中的智能体：成功取决于协调外部自然语言知识与工具输出，从而产生可验证且符合策略的状态变更。在我们新增的领域 τ-**银行业务**中，我们对逼真的金融科技客户支持工作流程进行建模。在这些工作流程中，智能体必须浏览约 700 份相互关联的知识文档，同时执行由工具介导的账户更新。在基于嵌入的检索和基于终端的搜索任务中，即使是推理预算充足的前沿模型，pass^1 也仅为 ∼25.5%，且可靠性在重复试验中急剧下降。智能体难以从密集互联的知识库中检索正确文档，也难以准确推理复杂的内部策略。总之，τ-知识提供了一个逼真的测试平台，可用于开发能在面向人类的部署中整合非结构化知识的智能体。

1. 引言

如今，语言智能体常被部署在需要与私有、自定义及非结构化知识库（KB）交互的场景中。在此类部署中，智能体必须以这些知识来源为依据采取行动、回应人类用户，并发现内部能力以及支配其行动的规则和约束（Wang &Yuan，2025 年；Xu&Peng，2025 年）。尽管现实世界中这类场景十分普遍，现有的语言智能体评估基准却很少能体现其复杂性。大多数既有研究要么单独评估检索，通常用于问答或网页搜索（Thakur 等，2021；Muennighoff 等，2023；Su 等，2025），要么单独评估工具使用（Yao 等，2025；Li 等，2025a；Huanget al.，2024 年），而不要求智能体对包含分布外产品术语、详细政策和可发现能力的私有知识库进行推理。此外，在这些场景中，检索通常发生在与用户的实时交互过程中，因此会因目标未充分指定、查询含糊以及用户意图不断变化而产生显著的随机性。

为弥补这一空白，**我们提出 τ-Knowledge，即 τ-Bench（Yao 等，2025）的扩展，用于通过一个新领域评估智能体在知识锚定环境中的表现，τ-Banking**。 τ-Banking 是一个新的金融科技领域，其中任务成功取决于从约 700 篇涵盖产品详情、程序性政策和工具文档的自然语言知识库文档中查找并应用信息。为在保持内部一致性的同时大规模构建该语料库，我们采用结构化到非结构化生成流水线，将产品、政策和工具的潜在结构化规范扩展为自然语言文档。这些任务模拟现实的客户支持流程，例如开立和关闭账户、兑换推荐促销以及处理直接存款咨询，同时要求智能体在长程对话中协调知识库证据与工具输出。重要的是，正如现实场景部署中那样，智能体并不能完全观察到各项能力——工具仅在文档中被提及，因此必须被找到才能使用。这一设计确保，对状态变更操作的访问取决于正确的知识检索。

我们构建 τ-**Knowledge，使其不受检索机制的限制，从而能够评估多种搜索策略**。该基准支持并评估在语料库中搜索和交互的任意策略，包括密集检索和稀疏检索、长上下文处理、基于文件系统的探索，以及混合方法。

*同等贡献 [1]Sierra [2]普林斯顿大学。通讯作者：Quan Shi < ben.s@sierra.ai>。

预印本。工作进行中。

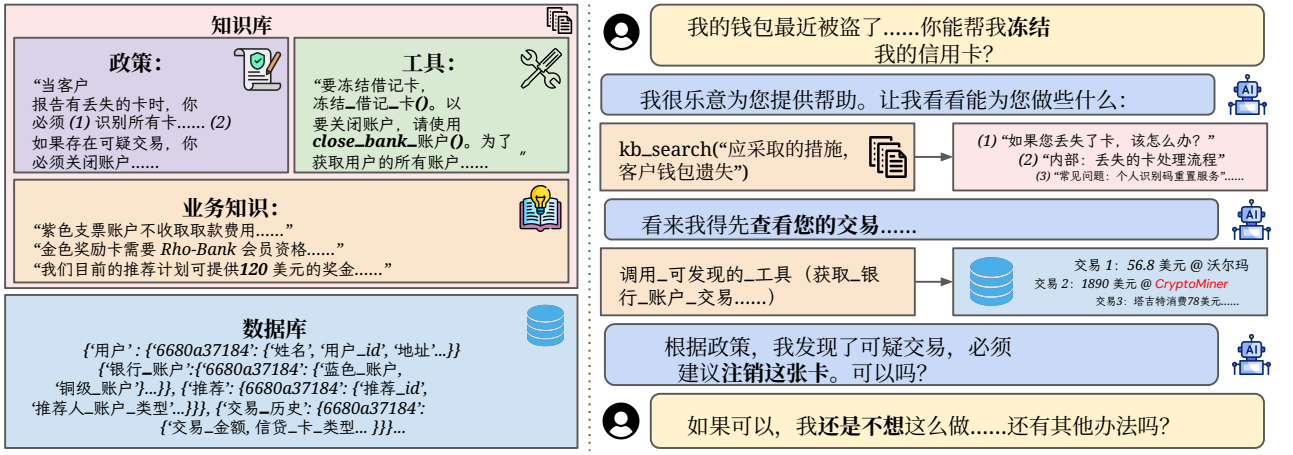

图1. τ-银行领域概览。智能体必须与知识库交互，以获取程序性知识、政策、工具和业务产品，从而通过调用发现的工具来修改底层数据库，解决复杂的用户请求。右侧示例展示了一个智能体如何协助一位丢失了装有银行卡的钱包的用户：尽管用户最初请求冻结银行卡，但针对特定银行卡的政策和交易历史限制要求智能体改为注销该卡。

这种灵活性使 τ-Knowledge 能够评估语义检索之外的新兴范式，例如通过终端在非结构化文档中导航。

我们的结果表明，**τ-Knowledge 能够有效揭示以知识为依据的智能体性能中的现实瓶颈**。在所有测试过的前沿智能体模型和检索配置中，观测到的最佳结果也仅为 25.52% pass^1（GPT-5.2，高推理），其中 pass^k 表示在 k 次独立试验中，任务每次都成功完成的概率。可靠性急剧下降，pass^4 最高仅为 13.40%。值得注意的是，即使通过黄金检索器设置消除了检索这一瓶颈，性能依然很低；在该设置中，智能体会直接在上下文中获得任务关键文档。此配置下最强模型的 pass^1 也仅为 39.69%（Claude-4.5-Opus，高推理）。这一差距表明，τ-Knowledge 并非仅靠检索就能解决，还需要智能体对复杂政策、跨文档依赖关系以及不断变化的数据库状态进行推理。

我们还发现**前沿模型与检索配置在性能和效率方面存在系统性差异**。例如，使用基于终端的搜索时，GPT-5.2 (high) 的性能与 Claude-4.5-Opus (high) 相当，但所需令牌数约多 1.7×，执行的 Shell 命令约多 2.3×，完成任务所需时间也约长 9×。一个常见原因是，当人类意图表述不明确时，搜索行为容易僵化并受假设驱动。有些模型不会通过澄清或定向检索来消除歧义，而是倾向于过度坚持早期假设，或在缺乏明确目标的搜索之间来回摇摆，导致交互轮次和工具调用过多。**检索配置会进一步放大这些差异**。例如，相较于基于嵌入的语义搜索，自由形式的基于终端的搜索可以提升推理能力较强的模型的性能，但通常需要显著更多的搜索步骤和工具交互。相比之下，稠密检索通常能带来更快的智能体轮次和更少的搜索。这些权衡表明，较弱或噪声较大的检索往往会通过增加搜索和工具使用来弥补，从而维持成功率，但代价是延迟更高、交互效率更低。

这些低效在面向人类的在线部署中影响尤为显著，因为额外的轮次会导致解决时间延长、认知负荷增加、信任度降低，在时间敏感型支持场景中尤其如此。因此，衡量面向人类的智能体的进展，不应只看最终任务成功与否，还应考察其 *解决方案效率*：以最短时间、最少工具调用和最少对话回溯，得出正确且符合政策的结果的能力。τ-Knowledge 为研究搜索、推理与效率在基于知识的智能体中如何相互作用提供了受控测试平台，并揭示了现有系统中存在的重大差距。

2. 相关工作

智能体与工具使用的基准 许多基准评估智能体将任务分解为多步计划、调用外部工具并执行结构化流程以实现预定义目标的能力（Jimenez 等，2024 年；Shi 等，2024 年；Xu 等，2024 年；Huang 等，2024 年；Wei 等，2025；Li 等，2025a；Mialon 等，2024 年）。然而，这些基准通常假设存在完全指定的工具接口，并评估孤立运行的智能体，而不对交互式用户进行明确建模

或对话动态。 τ-基准（Yao 等，2025；Bar- res 等，2025）通过引入面向目标的、部分可观测的对话环境来解决这些局限，在这些环境中，智能体必须在较长的时间范围内与模拟用户进行交互。尽管如此，在这些设置中，可用的工具和流程在很大程度上仍由智能体预先提供。 τ-知识直接建立在这一基础之上，要求智能体通过从自然语言语料库中进行检索来获取程序性知识，其中包括从文档中发现可用工具。

检索与以知识为中心的评估 大量既有研究通过跨领域的查询—文档匹配来评估嵌入质量（Thakur 等，2021；Muennighoff 等，2023；Sun 等，2024 年；Li 等，2025b；Wang 等，2024 年；Lin 等，2024 年；Song 等，2025；Su 等，2025）。尽管这类基准能有效衡量相关性，却无法反映知识获取如何影响决策、工具使用或长时域任务成功。其他研究将检索整合到面向任务或多轮问答中，以及序列化查询场景中（Katsis 等，2025；Cheng 等，2025；Kuo 等，2025；Mao 等，2024 年），但其中许多基准仍主要基于事实，几乎不需要对文档进行推理。相比之下，现实世界中的智能体会利用知识指导长时域推理与行动，通常依赖长上下文处理、模式匹配或混合策略，而非仅依靠稠密检索。其他基准（Dou 等，2026 年）评估智能体对长上下文进行推理并加以运用的能力，但测试场景并非对话环境，也不包含搜索／检索组件。τ-知识通过将知识获取抽象为与自然语言语料库的交互，并根据知识使用对任务完成和可靠性的影响来评估知识使用，从而弥补上述不足，在单一框架中统一基于检索、长上下文和工具增强的方法。

模拟人类与智能体的交互 模拟人类行为正日益用于评估和训练交互式智能体。既有工作包括基于人物设定的模拟器（Shi 等，2025；Park 等，2024 年）、教育场景中的人为错误模拟（Ross 与 Andreas，2025），以及目标导向的人机交互（Philipov 等，2024）。尽管这些方法提升了真实感，但许多用户模拟器会通过提示无意中向智能体透露未来的对话状态或结果，实际上充当了不知情的预言者。与先前的 Tau-Bench 用户模拟器（Yao 等，2025；Barres等人，2025）一样，τ-Knowledge 使用基于流程的用户模拟，并以当前环境状态为条件。此外，它还整合了可通过知识库发现的用户工具，使智能体能够在共享环境中将行动委托给模拟用户，并在不暴露有关未来状态的特权信息的情况下实现指令遵循。

3. τ-知识

τ-知识引入了一个新领域 τ-银行，要求智能体在正确行为取决于访问并应用外部知识库信息的场景中满足用户请求。

概述 每个任务在 τ-银行中模拟两名参与者之间真实的客户支持交互：一名 智能体和一名 用户 ，并可形式化为去中心化部分可观测马尔可夫决策过程（Dec-POMDP）（Bernstein 等人，2002）。智能体必须通过检索相关知识、依据政策和约束进行推理，并调用会修改底层银行数据库的工具，在多轮对话中帮助用户实现目标。

环境维护一个共享的全局状态 $S = S_{\text{db}} \times S_{\text{history}}$，由以下部分组成：(i) 银行数据库状态$S_{\text{db}}$，用于跟踪账户、交易、推荐和其他系统实体；(ii) 交互历史 S_{history}，用于记录智能体与用户之间的完整对话。智能体无法直接观察或修改 S_{db}；相反，它必须通过已记录的工具采取行动，并仅根据工具输出和用户消息推断状态。

动作由智能体调用工具（从其动作空间 A_{agent}中选择）构成；在某些情况下，也由用户执行（动作 A_{user}）。观察来自工具响应和对话轮次。由于信息和能力都要通过知识库获取，智能体必须先检索并正确解读关于政策、流程和可用工具的文档，才能执行有效的状态变更操作。

这种设计带来了部分可观测性：智能体和用户对 S 的了解不对称且不完整，因此任务成功可以客观验证：每项任务都会指定一个目标数据库状态，而任务奖励 $R: S \to [0,1]$取决于智能体检索知识、调用工具并进行交互后，是否在 S_{db}中生成了正确的最终状态。

知识库 如表 1所示， τ- 银行知识库包含 698 份文档，涵盖 21 个产品类别中的 71 个不同主题。覆盖范围包括个人支票账户和企业支票账户、分级储蓄账户（铜级至钻石精英级）、奖励信用卡、先买后付计划等。文档不仅详述面向客户的产品规格，如年收益率、最低余额、取款限额、现金返还方案、年费和加急运费，还包括内部客服人员操作规范：办理补发卡的流程（原因包括欺诈、丢失、

表1. 新 τ-银行领域的摘要统计。

统计量	τ-银行
知识库	
总词元数 *(cl100k base)*	194,562
# 文档	698
平均词元数/文档	278.7
# 知识类别	21
# 可发现工具	51
任务与交互	
# 永久代理工具	14
# 任务	97
平均每项任务所需文件数	18.6
平均每项任务所需工具调用次数	9.52
最少所需工具调用次数	1
最多所需工具调用次数	33
评估	
成功指标	pass^k

被盗或损坏）、账户关闭的资格要求、推荐计划规则和状态码、身份验证流程、联名账户持有人权限以及用户封锁政策。虽然数据库工具允许智能体直接观察 S_{db}，但语料库编码了如何正确根据该状态采取行动的规则。

可发现工具 在现实世界中，智能体能否找到说明如何执行操作的文档，会对系统状态产生明确且可验证的影响。为此，τ-Banking 引入了可发现工具：这些工具最初对智能体不可用，且仅在知识库中被隐式提及。可发现工具以函数签名的形式出现；要使用此类工具，智能体必须调用工具 us- ing call_discoverable_ tool(name, kwargs)。形式上，工具发现对应于智能体动作空间 A_{agent}的依赖状态的扩展，该扩展取决于智能体在 S_{history}中的知识状态。通过这种构造，能否访问特定的改变环境的操作取决于智能体能否获取相关知识，因此，知识获取失败会体现为系统状态的持续差异。

用户模拟 为实现可控且真实的评估，τ-Banking采用基于流程的用户模拟。每项任务都定义了一组条件规则，用于规定用户接下来采取的行动 a_{user}，这些行动取决于可观察到的智能体操作和/或环境结果：例如，如果智能体询问配送偏好，就要求加急配送；如果智能体冻结了三张借记卡，就透露其中一张其实是在夹克口袋里找到的。这让任务编写者能够精细控制评估的关键节点，引导对话进入边缘情况，测试智能体是否会正确拒绝不符合条件的请求，或在对话进行过程中引入需要适应的状态变化。 S_{db} 对话中未受明确流程规则约束的部分，则由基于大语言模型的用户模拟器自由生成，从而保留语言多样性和对话自然度。示例提示可见附录B。

4. 基准测试构建

我们通过一个多阶段流程生成知识库和基准任务，该流程结合了大语言模型生成与人工完善（Yao 等，2025；Barres等人，2025）。在整个流程中，我们在四种模型中交替使用大语言模型：GPT-5、GPT-5.2、Claude-4.5-Opus 和 Gemini-3-Pro，以使措辞、风格和文档结构更多样化。图2总结了知识库组件的创建过程。

阶段 1：结构化数据库生成 首先，我们使用大语言模型构建结构化知识库。此过程首先生成一组业务类别（例如信用卡、储蓄账户），接着为每个类别生成相关特征（例如卡片等级、账户协议），最后生成一组与每项特征相关的合理变量（例如年费、现金返现率、余额转账年利率）。最终得到一个结构化数据库，其中每项特征都表示为一组带有具体取值的带类型变量。从概念上讲，这个结构化知识库可视为一个约束系统：每个变量对应产品可能空间中的一个维度，而有效配置则对应于满足领域特定约束的可行赋值。

阶段 2：从结构化文档转换为非结构化文档 接下来，我们将结构化数据库转换为真实的非结构化文档语料库。对于每项特征，我们首先生成一组合理的文档标题（例如“铜级奖励信用卡概览”“如何查看我的每月现金返现？”）。然后，大语言模型将变量子集分配给可能包含这些变量的文档标题。最后，将每个文档标题及其相关变量交给大语言模型，由其生成一篇自然语言文章，对底层变量进行释义并补充上下文。此步骤将结构化数据库转换为类似真实客户服务文档的非结构化知识库，同时保留其与原始结构化表示的一致性。

阶段 3：任务与数据库创建 在初步构建知识库后，我们在大语言模型的协助下手动共同构建了任务和数据库，以反映金融科技客户服务中的常见流程，例如申请补办卡、对交易提出异议以及推荐账户。每项任务都围绕特定工作流程构建，并更新相应的知识文章和工具以

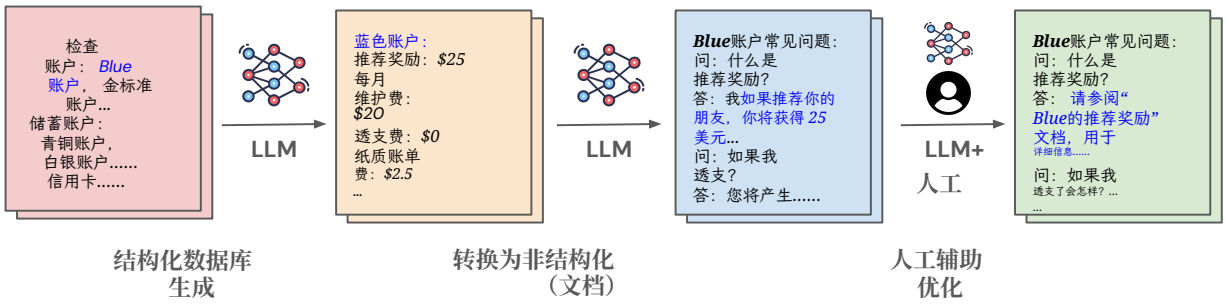

图 2. τ-Banking 的知识库构建流程。首先，大语言模型（LLM）将高层级产品／类别列表扩展为包含产品与服务及类型化属性（例如费用、奖励和限额）的结构化模式。然后，将结构化记录转换为自然语言文档（例如常见问题解答和政策文章），将底层变量分布到各文档中。最后，在任务创建期间，由人工和大语言模型审查、编辑、关联并去重内容，从而生成最终语料库。

支持该任务。例如，创建信用卡推荐任务需要向知识库添加概述推荐政策、符合条件的账户和奖励结构的文档。此外，每项任务都包含完成任务所需的最少文档清单（金标准文档）。任务示例见附录B。

阶段 4：人在回路中的优化 随着任务不断创建，我们通过添加、删除或修改变量来迭代优化结构化知识库，以满足新的任务需求。随后，我们仅针对知识库中受影响的部分，有选择地重新运行从结构化到非结构化的生成流程。此外，我们还进行了一些手动编辑，以提升内容的清晰度和真实性。完成这一流程后，我们得到一个大型非结构化知识库，其中包含约 700 篇文档和 200,000 个词元，涵盖 21 个产品类别中的数十个主题（见表1）。

阶段 5：审查 为确保任务正确无误，所有任务及其相关的金标准文档集均由两位未参与任务创建的审查者独立审核。对于每项任务，审查者均会确认：(i) 预期的最终数据库状态是否正确；(ii) 提供的金标准文档集是否完整且精简；以及 (iii) 仅使用金标准文档和已记录的工具能否成功完成任务。审查者会针对每项任务手动模拟至少一条有效执行轨迹，以确认任务在金标准条件下可解。完成大规模实验后，所有执行轨迹都会再次接受审核，以确保不存在非预期的捷径、规格漏洞或退化策略，使模型无需基于预期文档进行适当推理便能成功。此流程旨在减少基准测试伪影，并确保失败反映的是模型局限，而非任务设计错误。

该协议的优势 这一构建流程具有多项优势。(1) 它**高度可扩展**：大语言模型能够自动完成知识库创建的大部分阶段，同时人工干预仅限于有针对性的优化，这与许多长上下文基准中的做法一致（Bertsch 等人，2025；Hsieh 等人，2024 年）。(2) 它 **最大限度地减少意外冲突**。在生成过程中，每项功能都在结构化数据库中独立定义，只有下游任务需要时才引入功能之间的交互，而不是在文档生成过程中隐式产生。(3) 从结构化表示入手，可以大幅简化任务创建。每项任务都可以表示为针对知识库变量的一组约束（例如，APY 低于 3%、支持提前直接存款且不收取透支费用的唯一储蓄账户必须是 *Sky Blue* 账户）。这种表述方式有助于系统地设计和验证任务：可以直接根据结构化数据库验证约束，并通过单元式测试确保随着任务数量增长到数百个，新任务仍定义明确且互不冲突。

5. 实验

我们的实验旨在了解当前知识增强型智能体系统的能力。本节将介绍我们评估的检索／搜索方法和智能体模型。我们在附录 F 中进行了大量消融实验，以确保我们的实验配置反映当前最先进的水平。

稠密/稀疏检索 检索增强生成在企业应用中无处不在，因此自然成为我们评估的重点。我们评估稠密检索，它使用基于嵌入的语义相似度搜索；以及稀疏检索，它使用 BM25（Robert- son 等人，1995 年）。对于稠密检索，我们考虑了两个嵌入模型，它们因在大规模文本嵌入基准（MTEB）（Muennighoff 等人，2023 年）上的出色检索基准表现以及业界采用情况而入选：`text-embedding-3-large`（OpenAI，2024 年）和`Qwen3-embedding-8B`（Zhang 等人，2025b）。在所有检

索条件下，智能体都可以访问单个知识库_搜索工具，该工具返回排名前 k=10 的文档。智能体为该工具构造查询输入，并且每轮都可以重复使用该工具，包括在查看先前搜索调用的输出之后。对 k、嵌入模型以及是否存在逐点重排序器进行的消融实验显示，要么没有显著改进，要么性能下降；我们在附录F中报告了这些结果。

终端使用 现代 AI 辅助开发工具，例如 Cursor（Anysphere，2024 年）和 Claude Code（An-thropic，2025）通常通过基于终端的探索来浏览按代码库组织的知识，以更细粒度、可组合的探索方式取代单一的高级搜索原语。更广泛地说，终端正逐渐成为语言模型与非结构化信息交互的普遍界面（Merrill 等人，2026 年；Zhang 等人，2025a），因此我们在基准中将其纳入检索方式之一。我们的终端配置仿照 Terminal-Bench 中的 Terminus 智能体（Merrill 等人，2026 年），将知识库导出为沙箱文件系统中的文件，并为智能体提供一个用于执行任意 Unix 命令的 Shell 工具。智能体可以使用 `grep`、`cat` 和 `find` 等标准实用工具，按照自己的策略探索文档。详情请参阅附录D.2 和F。

黄金检索器 为了量化性能差距有多少归因于知识获取，而有多少归因于知识利用，我们还评估了一种“黄金检索器”配置：智能体在上下文中会收到完成任务所严格必需的全部文档。这组金标准文档在任务创建过程中整理而成，随后通过执行轨迹分析和人工检查进行验证。

上下文管理 由于智能体可能进行不受限制次数的检索调用，上下文长度可能迅速增长并超出模型的上下文窗口。为以尽量少的工程开销缓解上下文溢出，我们实现了一种轻量级截断策略。当对话超过模型的上下文限制时，我们会清除最早的知识检索调用所产生的输出（例如 `KB_search` 或 Shell 命令），删除迄今累积的所有检索输出中的四分之一。每个被删除的片段都会替换为一个占位符，提示智能体可以重新发起相应查询以恢复内容。实际上，截断很少发生：仅在 GPT-5.2 采用高推理力度时出现（该模型产生的检索调用数量最多；见第 6 节），且只发生在约 1–3% 的运行中。更复杂的上下文管理策略（例如选择性摘要或检索感知压缩）留待未来工作探索。

模型 我们评估了一组常用于面向消费者的应用的前沿语言模型。对于每家提供商，我们选择一个旨在最大限度提升推理性能的旗舰模型，以及一个专为更快交互而设计的低延迟变体。更快的变体要么是较小的同系列模型（如有，则采用默认推理力度），要么是降低推理力度配置下的同一模型。具体而言，我们评估了 GPT-5.2、Claude-4.5-Opus、Claude-4.5-Sonnet、Gemini-3-Pro 和 Gemini-3-Flash，这些模型均通过各自的企业级 API 调用。用户模拟器统一采用低推理力度的 GPT-5.2；我们发现它很少出现任务关键错误。

评估指标 与先前的研究类似（Barresetal.，2025；Yao 等，2025），我们使用 pass^k 指标评估智能体性能，该指标定义为任务在全部 k 次独立试验中均成功完成的概率。我们评估 $k \leq 4$。

6. 结果

表 2 报告了所有模型/配置的 pass^1 性能。以下总结了主要发现。完整结果（包括 pass^1–pass^4）、工具调用的详细分析以及统计显著性分析见附录G。

τ-知识带来显著挑战 表现最佳的配置是使用终端的 GPT-5.2 (high)，其 pass^1 仅为 25.52；紧随其后的是使用终端的 Claude-4.5-Opus 和 Claude-4.5-Sonnet，pass^1 分别为 24.74 和 22.42。随着 k 增加，性能急剧下降，最高 pass^4 也仅为 13.40（使用 Qwen3-emb-8b 的 GPT-5.2 (high)）。值得注意的是，即使直接在上下文中提供金标准文档，得分最高的模型（Claude-4.5-Opus）的 pass^1 也仅为 39.69，pass^4 则降至 26.80。这表明，在 τ-知识任务中，成功不仅取决于检索能力，还取决于仔细推理文档中动态且丰富的信息的能力。

可靠性与效率在不同模型间呈现分化 如图3所示，不同模型的可靠性下降速率各不相同。GPT-5.2 (high) 的整体 pass^4 得分最高（13.4）；而 Gemini-3-Flash 等模型尽管 pass^1 得分颇具竞争力，表现却下滑得更明显。将轨迹时长与性能一并分析，还能进一步揭示不同模型在解决方案效率上的差异。例如，Claude模型的性能与GPT模型相当，但完成任务所用时长显著更短。这既是因为Claude的token生成总量低于GPT（0.7M 对比 1.2M），也是因为Claude的工具调用更少，其中Claude-4.5-Opus发出的调用更少

表2. τ-知识的主要结果（pass^1，以 % 计）。各列表示不同的检索配置：text-embedding-3-large 和 Qwen3- emb-8b 使用各自的嵌入模型进行稠密检索；BM25 表示稀疏词法检索；终端通过文件形式提供对知识库的 Shell 访问；金标准将真实文档直接置于上下文中，从而在评估中移除检索环节。括号中的数值表示该行结果与金标准设置之间的差值。

模型	推理	Gold	嵌入 text-emb-3-large	嵌入 Qwen3-emb-8b	BM25	终端
GPT-5.2	High	32.73	**23.45** (-9.3)	**24.74** (-8.0)	**24.48** (-8.2)	**25.52** (-7.2)
GPT-5.2	None	15.72	8.25 (-7.5)	12.37 (-3.4)	9.54 (-6.2)	11.60 (-4.1)
Claude-4.5-Opus	High	**39.69**	18.30 (-21.4)	19.59 (-20.1)	17.78 (-21.9)	24.74 (-14.9)
Claude-4.5-Sonnet	High	33.76	17.53 (-16.2)	17.78 (-16.0)	16.75 (-17.0)	22.42 (-11.3)
Gemini-3-Pro	High	33.25	12.89 (-20.4)	12.89 (-20.4)	13.66 (-19.6)	15.72 (-17.5)
Gemini-3-Flash	High	36.34	18.56 (-17.8)	18.56 (-17.8)	18.56 (-17.8)	20.62 (-15.7)
平均值	—	32.18	16.88 (-15.3)	17.11 (-15.1)	17.04 (-15.1)	19.20 (-13.0)

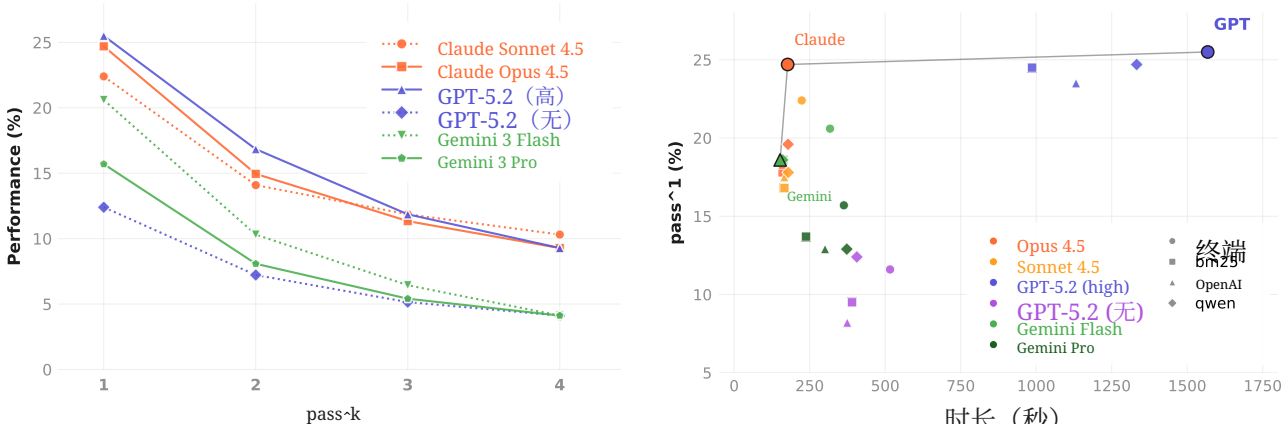

图3。（左）随着 k 增加，各模型表现最佳配置的 pass^1–4可靠性 急剧下降，不同系统之间的可靠性存在显著差异（例如 Gemini-3-flash 与 GPT-5.2 (high)）。（右）每项任务的平均耗时与 pass^1表现的帕累托前沿凸显了解决方案效率的显著差异。

检索（KB_搜索）每项任务平均调用次数为 8.7 次，相比之下，GPT-5.2高推理为 18.5 次。我们将在第7.2节讨论这种分化背后的定性原因。

近期模型受益于自由形式搜索 在各模型上取平均后，终端使用配置在统计显著性上优于密集检索和稀疏检索。然而，这一优势并非在所有模型类别中都一致。未启用推理的 GPT-5.2 不会从终端使用配置中受益，较早的 GPT 模型（GPT-4o 和 GPT-4.1；见附录 G）也同样如此。收益主要集中在近期的高推理模型上。这些结果表明，有效利用自由形式搜索——尤其是胜过结构化语义检索——依赖于只有最新一代推理优化模型才具备的能力。值得注意的是，这些改进并未伴随文档召回率的普遍提高（见下文的文档召回率分析），这表明收益并不仅仅来自检索到更多相关文档，还来自以更具策略性的方式对其进行推理和解析。

检索配置会影响解决任务所需的时间除了不同检索配置在 pass^1 分数上的差异外，我们还发现，不同检索方法的搜索频率存在显著差异：稠密检索平均每个任务搜索 9.9–10.1 次，而 BM25 为 11.4 次，终端使用中的 grep 为 14.5 次（终端使用配置下，所有 shell 调用的平均调用次数为每个任务 28.8 次）。因此，终端使用配置下的轮次耗时更长；与稠密检索配置相比，在所有模型组中，其轮次中位时间增加了 6.6 秒。在最慢轮次的第90百分位数处，差距进一步扩大：对于 Gemini-3-pro，终端使用比次慢策略约慢 10%；而对于 Gemini-3-



图4. τ-Bench 中具有代表性的智能体失效模式，分为四类：（1）复杂的产品相互依赖关系导致推荐错误；（2）因规划不足而违反必要的任务顺序；（3）过度信任用户陈述导致行动遗漏或未经核实；（4）检索驱动的决策过程中出现搜索低效和毫无根据的假设。

flash。这些发现表明，在搜索不受限制的智能体场景中，较弱的检索器往往可以通过提高搜索频率来弥补不足。虽然这有助于维持任务成功，但也可能延长交互时长并增加延迟；这些成本在面向人类的部署中不容忽视。

将搜索与信息使用区分开来为了更好地理解 τ-Knowledge 中失败的原因，我们进行了补充分析，以区分因搜索无效而导致的失败，以及因错误使用检索到的信息而导致的失败。首先，我们使用黄金检索器配置来单独评估推理表现，在这种配置下，金标准文档会直接提供在上下文中。即使不再需要搜索，通过率仍然很低，这表明模型难以正确解读和运用这些文档。其次，为了更准确地评估模型有效搜索的能力，我们采用文档召回率指标，其定义为在执行轨迹中至少一次出现在智能体上下文中的金标准文档所占百分比。尽管不同检索配置的平均召回率相近，但不同模型与检索器的配对之间差异显著。例如，text-embedding-3-large 与 Opus 配合时，文档召回率达到 57%；而 GPT-5.2 在不进行推理时使用它，召回率仅为 28%。这些差异表明，检索质量并非仅由底层检索器决定，还取决于智能体构造和迭代搜索查询的效率。

额外消融实验我们另外运行了两项基线测试，以验证我们的基准测试设计。首先，在**无知识**设置中，智能体无法访问知识库，并由默认策略引导。我们发现，所有模型的平均 pass^1 都降至仅 ~2%，这证实了这些任务确实需要检索到的信息。该数值并非零，是因为其中两项任务属于依据核查，用于验证信息缺失时智能体不会产生幻觉。其次，在**长上下文**设置中，我们将完整知识库直接附加到系统提示中，供上下文窗口足够大的模型（GPT-5.2、Gemini-3-Pro/Flash）使用，以检验我们的非金标准文档是否会引入有意义的混杂因素。表现最高也仅达到 pass^1 ~12%（GPT-5.2 (high)、Gemini-3-Pro），这表明额外文档成功制造了真实的噪声，而且即使技术上可以使用完整上下文，定向检索仍然有益。

7. 定性分析

为了更深入地了解总体性能指标之外智能体的失败模式，我们对

模型轨迹进行定性分析，以考察模型为何失败、交互过程中如何产生错误，以及失败究竟源于智能体推理、工具使用，还是用户模拟本身。

7.1. 用户模拟可靠性

与 τ-Bench 之前的迭代一样，用户由大语言模型模拟，这可能引入随机性和/或指令遵循错误，因此我们必须确认这些问题不会不公平地惩罚或奖励智能体。为量化模拟器的可靠性，我们从每项任务中随机抽取两条对话轨迹（来自不同的智能体模型），并请两位具备客户服务交互领域专业知识的标注员，将每条用户话语标注为 无错误、任务良性（不会影响任务可解性的轻微不一致），或 任务关键（即使智能体正确，也可能阻碍任务成功完成的行为，例如幻觉式核验细节、矛盾约束或目标偏离意图）。在标注的 194 条执行轨迹中，只有 4 条包含 任务关键的用户错误，因此关键错误率较低，与先前工作中的 τ-Telecom 领域相当（Barres等人，2025）。

7.2. 智能体错误分析

理解智能体错误模式对于解读基准结果和指导未来改进至关重要。因此，我们使用大语言模型分析所有受测配置中的失败执行轨迹，根据任务规范和模型执行轨迹生成失败解释，并对这些解释进行聚类，以识别常见错误模式。我们的发现概要见图 4。以下重点介绍四种最常见且最有参考价值的情况：

金融产品优惠之间复杂的相互依赖关系（∼14.5%） 知识型 τ面临的一项核心挑战是，产品和政策之间存在深层的相互依赖关系，因此需要跨多个文档进行多跳推理，才能找出最优解决方案。如图4 （左上，`tau_任务_008`）所示，该任务要求智能体在满足特定约束的前提下，最大限度地提高用户储蓄账户的年化收益率，并可在有利时开立其他产品。尽管一些文档宣传将储蓄账户与 Crypto Cash Back信用卡搭配可提升年化收益率，但这些奖励不如其他储蓄账户提供的较高基础年化收益率。许多智能体错误地将促销加成置于基础利率之上，尽管满足了表面上的激励条件，仍推荐了次优的产品组合。

未能遵守隐式子任务顺序（∼5%） 在 τ-知识中的某些任务涉及各项操作之间的隐式依赖，完成一项请求可能会阻断或使另一项失效。如图 4 （右上，`tau_-任务_053`）所示，用户既要求对一笔交易提出争议，又要求申请信用额度提升。然而，银行政策规定，若存在待处理争议，信用额度提升会被自动拒绝。因此，要成功完成，智能体必须推断并遵守隐式拓扑排序：先解决争议，再提交信用额度申请。许多智能体未能对这些依赖进行推理，而是按照用户给出的顺序执行操作。

过度信任用户陈述（∼4%） 在若干任务中，智能体未能根据系统状态核实用户提供的信息，而是直接照单全收，因而出错。如图4（左下，`tau_任务_027`）所示，智能体正确发起了一笔交易争议，并收到系统回复，表明该争议仍在审查中。随后，用户声称所有争议均已获批，并要求智能体发放相应贷记；许多智能体未经核实便照做了。

搜索低效与擅自假设（∼23%） 当用户请求中的信息不够明确时，智能体经常擅自做出没有依据的假设，而不是通过询问澄清或针对性搜索来消除歧义。如图 4 （右下角，`tau_任务_098`）所示，用户询问哪个账户提供最高的推荐奖励，却没有说明账户类型。许多智能体立即假定用户指的是信用卡，并推荐多种信用卡产品，尽管现有文档也涵盖其他账户类型的推荐计划。低效的搜索策略和由假设驱动的处理过程会显著损害用户体验（Wang 等，2025），而用户体验是与人类互动的智能体实现现实场景部署时至关重要的一环。

8. 结论

我们推出了 τ-知识，其中包括一个新领域 τ-银行，用于评估智能体在对话场景中执行任务的表现；这些任务要求智能体检索、推理并运用非结构化、非参数化的知识。我们发现，即使是采用最先进检索配置的顶尖大语言模型，也难以稳定可靠地完成任务；同时，我们强调，解决方案效率是未来以人为中心的智能体部署研究中一个至关重要的方向。

局限性与未来工作 (1) 我们的用户模拟较为简化，未能体现真实人类交互的诸多特征，例如用户专业水平的差异、口语化或本地化语言，以及语法不完善或含义模糊的输入。(2) 我们的时长和效率指标取决于当前 API 提供商的服务特性，而绝对延迟可以在其他部署环境中进一步优化。(3) 我们在完全开放的智能体式搜索机制下评估智能体，允许其不受限制地进行

知识查询，而许多现实世界的系统则采用单样本或少样本搜索约束；在此类限额下，我们预计检索配置对性能的影响会更加显著。(4) 我们的终端使用场景值得进一步深入研究，尤其是写入工具的作用，因为设计更周密的智能体可能会更有效地利用笔记记录、状态跟踪和知识重组来提升长时域推理能力。

9. 致谢

我们感谢 Keshav Dhandhania、SohamRay、Siyu Yao 和 Vijay Iyengar 富有成效的讨论，这些讨论帮助我们完善了这项工作；也感谢 Clay Bavor 一如既往的支持。

10. 影响声明

我们预计这项工作不会带来直接的负面社会影响。相反，我们认为，τ-知识有助于推动人工智能系统的评估转向能够反映真实世界中人类结果的环境；在这些环境中，智能体必须通过互动、澄清和符合策略的决策来解决用户目标，而不能仅依赖自主任务完成。以往许多人工智能评测工作都侧重于完全自动化、无人参与的场景，因此忽视了系统与人互动时实际可能出现的各种问题，例如幻觉、不安全行为、互动效率低下或耗时过长，以及脆弱的决策。通过用户模拟明确建模人类行为，这一基准鼓励开发更可靠、透明且符合人类需求的人工智能系统。

参考文献

Anthropic. Claude code：Agentic coding，2025。访问时间：2025。Anysphere。Cursor：AI代码编辑器。https://cursor.com，2024 年。访问时间：2025。Barres, V.、Dong, H.、Ray, S.、Si, X. 与 Narasimhan, K. τ^2-bench：在双控环境中评估对话智能体。 *arXiv* 预印本 *arXiv:2506.07982*，2025。Bernstein, D. S.、Givan, R.、Immerman, N. 与 Zilberstein, S.。马尔可夫决策过程分散控制的复杂性。《运筹学数学》，27(4):819–840，2002。Bertsch, A.、Pratapa, A.、Mitamura, T.、Neubig, G. 与 Gormley, M. R.。Oolong：评估长上下文推理与聚合能力。 *arXiv* 预印本 *arXiv:2511.02817*，2025。Cheng, Y., Mao, K., Zhao, Z., Dong, G., Qian, H., Wu, Y., Sakai, T., Wen, J.-R., and Dou, Z. CORAL：多轮对话检索增强生成的基准测试。载于计算语言学协会研究成果：*NAACL 2025*，第 1308–1330 页，2025 年。Dou, S., Zhang, M., Yin, Z., Huang, C., Shen, Y., Wang, J., Chen, J., Ni, Y., Ye, J., Zhang, C., 等。CL-bench：上下文学习基准。*arXiv* 预印本 *arXiv:2602.03587*，2026 年。Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Z hang, Y., and Ginsburg, B. Ruler：你的长上下文语言模型的实际上下文长度是多少？*arXiv* 预印本 *arXiv:2404.06654*，2024 年。Huang, S., Zhong, W., Lu, J., Zhu, Q., Gao, J., Liu, W., Hou, Y., Zeng, X., Wang, Y., Shang, L., Jiang, X., Xu, R., and Liu, Q. 规划、创建、使用：针对真实复杂场景中大语言模型综合工具利用能力的基准测试。载于 Ku, L.-W., Martins, A., and Srikumar, V.（编），计算语言学协会研究成果：*ACL 2024*，第 4363–4400 页，泰国曼谷，2024 年 8 月。计算语言学协会。doi: 10.18653/v1/2024.findings-acl.259。Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. SWE-bench：语言模型能否解决真实世界的 GitHub 问题？载于国际学习表征会议（*ICLR*），2024 年。Katsis, Y., Rosenthal, S., Fadnis, K., Gunasekara, C., Lee, Y.-S., Popa, L., Shah, V., Zhu, H., Contractor, D., and Danilevsky, M. MTRAG：用于评估检索增强生成系统的多轮对话基准。计算语言学协会汇刊，13:784–808，2025 年。Koehn, P. 机器翻译评估的统计显著性检验。载于*2004*年自然语言处理实证方法会议论文集，第 388–395 页，西班牙巴塞罗那，2004 年 7 月。计算语言学协会。Kuo, T.-L., Liao, F.-T., Hsieh, M.-W., Chang, F.-C., Hsu, P.-C., and Shiu, D.-S. RAD-Bench：评估大语言模型在检索增强对话中的能力。载于*2025*年美洲计算语言学协会北美分会会议论文集：人类语言技术（产业专题），第 868–902 页，美国新墨西哥州阿尔伯克基，2025 年 4 月。计算语言学协会。doi: 10.18653/v1/2025.naacl-industry.66。Li, J., Zhao, W., Zhao, J., Zeng, W., Wu, H., Wang, X., Ge, R., Cao, Y., Huang, Y., Liu, W., Liu, J., Su, Z., Guo, Y., Zhou, F., Zhang, L., Michelini, J., Wang, X.,

Yue, X., Zhou, S., Neubig, G., and He, J. 工具十项全能：针对多样、真实且长周期任务执行的语言智能体基准测试。*arXiv* 预印本 *arXiv:2510.25726*，2025a。

Li, X., Dong, K., Lee, Y. Q., Xia, W., Zhang, H., Dai, X., Wang, Y., and Tang, R. CoIR：一个面向代码信息检索模型的综合基准。收录于*2025* 年计算语言学协会年会论文集。计算语言学协会，2025b。

Lin, C.-Y. ROUGE：一个用于自动评估摘要的工具包。收录于文本摘要拓展，第 74–81 页，西班牙巴塞罗那，2004年7月。计算语言学协会。

Lin, H., Zhan, S., Su, J., Zheng, H., and Wang, H. IRSC：一个用于检索增强生成场景下通过语义理解评估信息检索的零样本评估基准。*arXiv*，2409.15763，2024 年。

Long, M., Sun, D., Yang, D., Wang, J., Luo, Y., Shen, Y., Wang, J., Zhou, H., Guo, C., Wei, P., Wang, J., and Gu, J. DIVER：一种面向推理密集型信息检索的多阶段方法。*arXiv* 预印本 *arXiv:2508.07995*，2025。

Mao, K.、Deng, C.、Chen, H.、Mo, F.、Liu, Z.、Sakai, T.与Dou, Z. ChatRetriever：面向通用且稳健的对话式稠密检索适配大语言模型。载于*2024* 年自然语言处理经验方法会议论文集，第1227–1240页，美国佛罗里达州迈阿密，2024年11月。计算语言学协会。doi: 10.18653/v1/2024.emnlp-main.71.

Merrill, M. A.、Shaw, A. G.、Carlini, N.、Li, B.、Raj, H.、Bercovich, I.、Shi, L.、Shin, J. Y.、Walshe, T.、Buchanan, E. K.等。Terminal-bench：针对命令行界面中困难且真实任务的智能体基准评测。*arXiv*预印本 *arXiv:2601.11868*，2026 年。

Mialon, G.、Fourrier, C.、Swift, C.、Wolf, T.、LeCun, Y. 和 Scialom, T. GAIA：通用人工智能助手基准。见国际学习表征会议（*ICLR*），2024 年。

Muennighoff, N.、Tazi, N.、Magne, L. 和 Reimers, N. MTEB：大规模文本嵌入基准。见 第*17* 届欧洲计算语言学协会分会会议论文集，第 2014–2037 页，2023。

OpenAI。OpenAI API：嵌入模型（text-embedding- 3-large）。
h
t
t
ps://platform.openai.com/docs/models/text-embedding-3-large, 2024 年。访问于 2026-01-22。

Park, J. S., Zou, C. Q., Shaw, A., Hill, B. M., Cai, C., Morris, M. R., Willer, R., Liang, P. 和 Bernstein, M. S. 对 1,000 人进行生成式智能体模拟。*arXiv* 预印本 *arXiv:2411.10109*，2024 年。Philipov, D., Dongre, V., Tur, G. 和 Hakkani-T¨ur, D. 用于具身对话式 AI 的用户智能体模拟。*arXiv* 预印本 *arXiv:2410.23535*，2024 年。

Robertson, S.、Walker, S.、Jones, S.、Hancock-Beaulieu, M. 和 Gatford, M. Okapi 在 TREC-3。见第三届文本检索会议（*TREC-3*）论文集，第 109–126 页。美国国家标准与技术研究院，1995。

Ross, A. 和 Andreas, J. 学会制造 MISTAKEs：对错误的学生思维与关键错误进行建模。*arXiv* 预印本 *arXiv:2510.11502*，2025。

Shi, Q., Tang, M., Narasimhan, K., and Yao, S. 语言模型能解决奥林匹克编程问题吗？*arXiv* 预印本 *arXiv:2404.10952*，2024 年。

Shi, Q., Jimenez, C. E., Dong, S., Seo, B., Yao, C., Kelch, A., andNarasimhan, K. Impersona：评估个体层面的语言模型身份模仿。*arXiv* 预印本 *arXiv:2504.04332*，2025。

Song, T., Gan, G., Shang, M., and Zhao, Y. IFIR：用于评估专家领域信息检索中指令遵循能力的综合基准。计算语言学协会美洲分会 *2025* 年会议论文集：人类语言技术，2025。

Su, H., Yen, H., Xia, M., Shi, W., Muennighoff, N., yu Wang, H., Liu, H., Shi, Q., Siegel, Z. S., Tang, M., Sun, R., Yoon, J., Arik, S. O., Chen, D., and Yu, T. Bright：面向推理密集型检索的真实且富有挑战性的基准。发表于国际学习表征会议 *(ICLR)*，2025。

Sun, W., Shi, Z., Long, W. J., Yan, L., Ma, X., Liu, Y., Cao, M., Yin, D., 和 Ren, Z. MAIR：用于评估指令式检索的大规模基准。载于*2024*年自然语言处理实证方法会议论文集。计算语言学协会，2024 年。doi: 10.18653/v1/2024.emnlp-main.778。

Thakur, N., Reimers, N., R¨uckl´e, A., Srivastava, A., 和 Gurevych, I. BEIR：用于信息检索模型零样本评估的异构基准。*arXiv* 预印本 *arXiv:2104.08663*，2021 年。

Wang, J., Ming, Y., Dulepet, R., Chen, Q., Xu, A., Ke, Z., Sala, F., Albarghouthi, A., Xiong, C., and Joty, S. LiveResearchBench：面向真实场景中以用户为中心的深度研究的实时基准。*arXiv* 预印本 *arXiv:2510.14240*，2025。

Wang, X., Wang, J., Cao, W., Wang, K., Paturi, R., and Bergen, L. BIRCO：包含复杂目标的信息检索任务基准。*arXiv* 预印本 *arXiv:2402.14151*，2024 年。

Wang, Z. 和 Yuan, B. L-MARS：采用编排式推理和智能体式搜索的法律多智能体工作流。*arXiv* 预印本 *arXiv:2509.00761*，2025 年。Wei, J.、Sun, Z.、Papay, S.、McKinney, S.、Han, J.、Fulford, I.、Chung, H. W.、Passos, A. T.、Fedus, W. 和 Glaese, A. Browsecomp：面向浏览智能体的简单但富有挑战性的基准。*arXiv* 预印本 *arXiv:2504.12516*，2025 年。Xu, F. F.、Song, Y.、Li, B.、Tang, Y.、Jain, K.、Bao, M.、Wang, Z. Z.、Zhou, X.、Guo, Z.、Cao, M.、Yang, M.、Lu, H. Y.、Martin, A.、Su, Z.、Maben, L.、Mehta, R.、Chi, W.、Jang, L.、Xie, Y.、Zhou, S. 和 Neubig, G. TheAgentCompany：对大语言模型智能体执行现实世界关键任务的基准测试。*arXiv* 预印本 *arXiv:2412.14161*，2024 年。Xu, R. 和 Peng, J. 深度研究的综合综述：系统、方法论与应用。*arXiv* 预印本 *arXiv:2506.12594*，2025 年。Yao, S.、Shinn, N.、Razavi, P. 和 Narasimhan, K. τ-bench：现实世界领域中工具—智能体—用户交互的基准。在国际学习表征会议（*ICLR*），2025 年。Zhang, A. L.、Kraska, T. 和 Khattab, O. 递归语言模型。*arXiv* 预印本 *arXiv:2512.24601*，2025a。Zhang, Y.、Li, M.、Long, D.、Zhang, X.、Lin, H.、Yang, B.、Xie, P.、Yang, A.、Liu, D.、Lin, J.、Huang, F. 和 Zhou, J. Qwen3嵌入：通过基础模型推进文本嵌入和重排序。*arXiv* 预印本 *arXiv:2506.05176*，2025b。

A. 知识库文档示例

The τ-银行知识 该基础由 698 份文档组成。我们展示了三个关键文档类别的代表性示例 t 类型，智能体必须检索 并在任务执行过程中进行推理。

请注意，在构建阶段，文档可能包含模板变量（显示为 [[变量_姓名]]），这些变量会在基准最终确定时填入具体值。智能体必须检索这些文档，以回答产品比较问题、核实资格，或向客户解释条款。

A.1. 程序性知识（政策文档）

政策文档规定了智能体在处理客户请求时必须遵循的程序逻辑。这些文档包含分步规程、依据客户或账户状态进行的条件分支，以及执行各项操作所需的具体工具的相关说明。以下示例展示了信用卡挽留规程，该规程指导智能体核实资格、检查滥用记录、了解客户关切并提供适当的挽留优惠。

内部：信用卡挽留规程（节选）

步骤 1：核实销户资格
Bef更多尝试 i挽留 i上，con fi确认客户是否符合销户条件。按以下顺序检查：

1. **待处理争议：** 账户不得存在任何有效或待处理的交易争议。如有，请告知客户等待争议解决后再销户。
2. **没有待处理的补发卡：** 如果已订购补办卡，但尚未收到或激活，则无法继续销户。
3. **账户最低存续时间：** 账户必须至少开立 [[cc_销户_最低_账户_存续时间_天]] 天。若未达到此期限，请告知客户目前还不能销户。
4. **未偿余额：** 账户不得有任何未偿余额 .

步骤 2：检查此前的挽留尝试U
se th获取_销户_原因_历史记录_i
_8293 工具，用于确定该账户在过去一年内是否有任何销户原因记录。如果有记录，则跳过挽留优惠，直接处理销户。工具参数：

- 信贷_卡_账户_ID（字符串，必填）：信用卡账户 ID。

步骤 3：了解并记录原因
询问客户想要销户的原因 使用 tool credit_card_closure_reason_4521 记录。工具参数：

- credit_card_account_id（字符串，必填）
- user_id（字符串，必填）
- closure_reason（字符串，必填）：以下选项之一：'年费'，'不再使用信用卡'，'找到更好的信用卡'，'对奖励不满意'，'简化财务'，'负面体验'，'其他'。

第 4 步：处理顾虑关闭
之后ilored solutions based 原因：

- **年费顾虑 卡 -_账户_标记_ ：** 如果客户持有时长达到 ≥ 2 年，可使用 apply_信贷_卡，提供免除一年年费的方案 6147。否则，提供永久降级为免年费信用卡的方案。
- **未使用这张卡：** 提醒客户卡片的福利；建议设置定期订阅 .
- **发现了更好的卡：** 询问是什么功能吸引了他们 . 如果 Rho-Bank 提供类似的卡片，请改为帮助他们申请 .

A.2. 可发现工具文档

可发现工具是特权操作，智能体无法立即使用。其文档嵌入在知识库中，智能体必须先找到该文档，然后才能解锁并调用工具。下面的示例展示了工具签名和调用模式在文档中的呈现方式。可发现工具

引入了一项关键的检索依赖：如果智能体无法找到相关文档，就无法执行所需操作。这种设计反映了现实场景：智能体的能力取决于其可访问的文档，而非硬编码的接口。

应用已解决的返现争议更正（内部）

概述
返现争议解决并获批后，您必须将受影响的交易更新为正确的奖励值。**必需步骤**

1) 找出需要更正的交易

- 在 cash_back_disputes 数据库中查找用户已解决的争议，以 find 需要调整奖励的交易_id 值。

2) 解锁内部更新工具

- 使用解锁_可发现的_智能体_工具来解锁智能体工具更新_交易_奖励_3847。

3) 重新计算正确的奖励

- Recalculate th正确的t 奖励 ds based 在车上d type, t_{ransac} $ti^{关于}_{ca}$ $t^{egory,\ 一}_{个}$ d applicable 促销 tions.
- 重要：请勿依赖争议记录中任何预期的_奖励字段。请独立核实利率和资格。

4) 应用更正后的奖励

- 使用可发现的_智能体_工具，通过以下信息调用更新_交易_奖励_3847：– 交易 ID：准确的交易标识符_– 新获得的_奖励_：格式为“X 积分”的字符串，其中 X 为正确的整数积分值

合规性与准确性

- 确保计算反映正确的基础费率，以及购买时生效的任何奖励类别或促销活动。
- 工具调用完成后，请在信贷_卡_交易_历史记录中确认更新。

A.3. 产品与业务变量文档

产品文件介绍各类银行产品与服务的功能、费用、利率和资格要求。

绿色账户（支票账户）一览

亮点

- 不收取透支费用，也不提供透支保障；超出可用余额的交易将被拒绝
- 支票账户余额可享受 [[apy_rate]]% 年收益率
- 直接存款最多可提前 [[early_direct_deposit_days]] day(s)
- 移动支票存款每日最高可达 $ [[mobile_deposit_limit_daily]] per day
- 提升关联储蓄账户的年收益率：金标准 +[[gold_savings_apy_boost]]% 或白银级 +[[silver_savings_- apy_boost]]%

费用和限额

Item	金额
月度维护费	$ [[每月_维护_费]]
免收费用的最低每日余额	$ [[最低_余额_以_免除_费用]]
纸质账单费（每月）	$ [[纸质_账单_费]]
非本网络ATM取款费用	$ [[ATM_取款_费用_非_本_网络]]
余额年收益率	[[年收益率_利率]]%
每日移动支票存款限额	$ [[移动_存款_限额_每日]]
提前直接存款	最多提前 [[提前_直接_存款_提前天数]] 提前天数

推荐计划

- 每成功推荐一位用户，您可获得：$[referral_bonus_amount]
- 他们获得：$[推荐_被推荐_奖金_金额]欢迎奖金
- 每年最多推荐次数：[推荐_奖金_最大_每_年]
- 需要符合条件的存款：$[推荐_奖金_符合条件_存款] 在 [推荐_奖金_- 符合条件_期限_天] 天

重要详情

- 要免除月度维护费，请在每个账单周期内维持不低于$[最低每日余额_余额_以_免除_费用]
- 提前直接存款到账时间取决于付款方提交存款的时间
- 关联符合条件的黄金级或白银储蓄账户后，即可享受储蓄账户年化百分比收益率提升

B. 任务示例

我们展示了来自 τ-Banking 的三个代表性任务，以说明知识驱动的智能体评估有多复杂。每个示例都包含用户模拟器提示（提供给模拟用户的指令）以及构成成功解决任务所需的预期操作序列。与以往单独测试检索或工具使用的基准不同，τ-Banking 任务要求智能体协同完成知识获取、约束推理和多步骤执行。

B.1. 示例 1：多约束产品推荐

此任务要求智能体根据对话中逐步透露的多项条件，对商业支票账户（8 个候选账户）和企业储蓄账户（7 个候选账户）进行筛选。排除不符合任何一项条件的账户后，支票账户和储蓄账户分别各有三个仍符合要求。随后，智能体必须在知识库中查找促销政策 以进一步筛选。关键在于，知识库中有四则促销通知——每种账户类型各两则，日期范围各不相同。智能体必须确定当前模拟日期（11/14/2025），并辨别哪些促销活动仍有效、哪些已经过期。十月促销（于 11/12 结束）优先推荐的账户与十一月促销不同。

用户模拟器提示

你的角色： 你是 Yumi Tanaka，35 岁，是一家名为“Aurora Design Co”的小型创意设计工作室的老板，公司位于俄勒冈州波特兰。你在 Rho-Bank 开设 Green账户个人账户已有约 3 年，账户中大约有 3,150 美元。大约 2 个月前，你还为工作室开设了一个 Cobalt Blue 商业支票账户。

您的目标： 分别为您推荐一个新的商业支票账户和一个商业储蓄账户，并且两者都满足您的所有要求。您希望智能体为每种账户只推荐一个选项，而不是提供多个选项供您选择。**身份验证信息：** 姓名：Yumi Tanaka — 电话：503-555-0741 — 电子邮箱：yumi.tanaka@aurora.io 出生日期：08/23/1990 — 地址：2847 Willamette Street, 俄勒冈州波特兰 97202

对话流程：
开场白：“你好！我经营一家小型创意设计工作室，想开一个功能比我目前使用的 Cobalt Blue 更好的新商业支票账户，也想开一个商业储蓄账户来开始积累储备资金。我对这两种账户都有具体要求，希望你能直接告诉我哪些账户最适合我——我不想比较一大堆选项。”

自然地说明支票账户的要求：
• “我经常收到客户开来的大额支票——有时一次就是 8,000 美元或 10,000 美元。我需要一个账户，让我每天至少可以通过移动存款存入 10,000 美元。” • “我绝对不能支付透支费用。绝不收取透支费用，这一点没有商量余地。” • “我经营的是一家小型工作室，不是什么大公司。最低余额必须低于一万美元。” • “我也希望能赚取不错的利息——年收益率至少要有 1%。”显示储蓄账户要求：• “我需要当天到账的 ACH转账来管理现金流。” • “我无法承诺在储蓄账户中保留 50,000 美元或更多。” • “如果我需要电汇资金，我不想每笔电汇支付超过 15 美元的费用。” 如果有多个选项：“听着，我真的不想比较各种选项。根据我刚才说的所有要求，哪一个支票账户和哪一个储蓄账户最适合我？”

预期操作序列

1. 记录_身份验证（姓名=“Yumi Tanaka”，用户_ID=“yt71c9e4f2”，……） 2. 解锁_可发现的_智能体_工具（智能体_工具_名称=“获取_所有_用户_账户_依据_用户_ID_3847”） 3. 调用_可发现的_智能体_工具（智能体_工具_名称=“获取_所有_用户_账户_依据_用户_ID_3847”，参数={“用户_ID”: “yt71c9e4f2”}） 4. 解锁 _ 可发现的 _ 智能体 _ 工具（智能体 _ 工具 _ 名称 =“ 开立 _ 银行 _ 账户 _4821”） 5. 调用_可发现的_智能体_工具（智能体_工具_名称=“开立_银行_账户_4821”，参数={“用户_ID”: “yt71c9e4f2”, “账户_类型”: “商业_支票账户”, “账户_类别”: “Sky Blue”}） 6. 调用_可发现的_智能体_工具（智能体_工具_名称=“开立_银行_账户_4821”，参数={“用户_ID”: “yt71c9e4f2”, “账户_类型”: “商业_储蓄账户”, “账户_类别”: “Gold Saver Account”}）

B.2. 示例 2：程序性政策执行

此任务旨在测试智能体能否准确遵循知识库中记录的多步骤信用卡挽留规程。该规程要求：(1) 通过检查待处理争议、待处理的补发卡、账户最低存续时间和未偿余额，核实销户资格。智能体发现有 75 美元余额，必须先付清；(2) 使用 get_closure_reason_history_8293 检查过去一年内是否有此前的挽留尝试。如果有，则跳过挽留优惠；(3) 询问客户销户的原因并记录该原因；(4) 根据具体原因解决客户的顾虑。对于“年费”相关顾虑，政策规定，应向账户存续 2+ 年的客户提供年费豁免。智能体必须发现账户于 2022 年 11 月 10 日开立（表明客户已开户 3 年），并认识到该客户符合忠诚客户福利的资格，然后设置 annual_fee_waived 标记，并填写正确的到期日期。未执行任何一个步骤，或未按顺序执行步骤，都会导致任务失败。

用户模拟器提示

你的角色：你是 Yuki Nakamura，一位 33 岁的营销经理，来自俄勒冈州波特兰。你持有 Rho-Bank 的白金奖励信用卡，已经好几年了。**你的情况：**你一直在考虑注销白金奖励信用卡，因为最近使用这张卡的频率不高，年费对你来说实在太高了。**你的目标：**表达你想因年费而注销信用卡。如果智能体提到你还有一笔未偿余额需要先还清，请同意从你的支票账户中还清。如果智能体之后提出为你豁免一年的年费，请接受这个提议并保留信用卡——不要主动要求豁免，因为你不知道有这种选择。**身份验证信息：**姓名：Yuki Nakamura — 电话：503-555-0842 — 电子邮箱：yuki.nakamura@simba.com 出生日期：05/12/1991开场白：“你好，我想注销我的白金奖励信用卡。年费现在对我来说实在太贵了。” 被告知有未偿余额时：“哦，我不知道卡上还有余额。好的，请从我的支票账户中把它还清。” 付款处理完毕后：“太好了，谢谢你处理这件事。那我们现在可以继续注销信用卡了吗？” 被问到为何想注销时：“其实就是因为年费。以我现在使用这张卡的频率来说，年费太高了。” 如果对方提出豁免年费：“哦，你们可以豁免年费吗？那太好了！好的，麻烦你帮我办理。如果能豁免一年的年费，我很愿意继续保留这张卡。”

预期操作序列

1. 记录_身份验证(姓名="Yuki Nakamura", 用户_id="224959b99e", ...)

2. 解锁 di_ 可发现_智能体_工具(智能体_工具_姓名="获取_用户_争议_历史记录_7291")

```
3. 调用_可发现的_智能体_工具(                                    # 步骤 1：检查待处理争议
     智能体_工具_姓名="获取_用户_争议_历史记录_7291",
     参数={"用户_ID":                "224959b99e"})

4. 解锁_可发现的_智能体_工具(
     智能体_工具_姓名="获取_待处理_替换_订单_5765")

5. 调用_可发现的_智能体_工具(                                    # 步骤 1：检查待处理的替换
     智能体_工具_姓名="获取_待处理_替换_订单_5765",
     参数={"信贷_卡_账户_编号": "cc_224959b99e_plat"})

6. 解锁_可发现的_智能体_工具(
     智能体_工具_姓名="get_all_用户_账户_by_用户_id_3847")

7. 调用_可发现的_智能体_工具(                                    # 查找未偿余额（$75）
     智能体_工具_姓名="get_all_用户_账户_by_用户_id_3847",
     arguments={"用户_id":       "224959b99e"})

8. 解锁_可发现的_智能体_工具(
     智能体_工具_姓名="支付_信贷_卡_，从_支票账户_9182")

9. 调用_可发现的_智能体_工具(                                    # 步骤 1：先还清余额
     智能体_工具_姓名="支付_信贷_卡_，从_支票账户_9182",
     参数={"用户_id":                "224959b99e",
                  "支票_账户_id": "05",
                  "信贷_卡_账户_id": "cc_224959b99e_plat",
                  "金额":         75.0})

10. 解锁_可发现_智能体_工具(智能体_工具_姓名="获取_销户_原因_历史记录_8293")

11. 调用_可发现_智能体_工具(                                    # 步骤 2：检查之前的尝试
     智能体_工具_姓名="获取_销户_原因_历史记录_8293",
     参数={"信贷_卡_账户_ID": "cc_224959b99e_plat"})

12. 解锁_可发现_智能体_工具(
     智能体_工具_姓名="记录_信贷_卡_销户_原因_4521")

13. 调用_可发现的_智能体_工具(                                   # 步骤 3：记录原因
     智能体_工具_姓名="记录_信贷_卡_销户_原因_4521",
     参数={"信贷_卡_账户_编号": "cc_224959b99e_plat",
                 "用户 _id":  "224959b99e" ，
                 "销户      _原因":       "年度_费用"})

14. 解锁_可发现的_智能体_工具(
     智能体_工具_姓名="申请_信贷_卡_账户_标记_6147")

15. call_discoverable_智能体_工具(                               # 第 4 步：申请费用减免
     智能体_工具_姓名="apply_信贷_card_账户_flag_6147",
     参数={"信贷_卡_账户_id": "cc_224959b99e_plat",
                 "用户 _id":  "224959b99e" ，
                 "标记_类型": "年费_已_免除",
                 "有效期_日期":              "2026 年11/14",
                 "原因":       "忠诚度_优惠"})
```

B.3. 示例 3：带依赖关系的操作排序

此任务包含四项存在隐藏顺序依赖的用户请求。若按用户要求的顺序执行，将导致彻底失败：先关闭任何账户都会使其进入已关闭状态，从而无法办理商业支票账户（政策规定：“状态为 CLOSED 的账户一律不予办理”）；关闭 Evergreen 支票账户后，则只剩下一个开户仅 12 天的账户，未达到开设储蓄账户所需的 14 天账户存续期要求。智能体必须从不同的知识库文档中识别这些依赖关系，重新排列全部四项操作，向用户解释这些限制，并按正确顺序执行 12 次工具调用。这种拓扑排序要求智能体推理当前操作会如何限制未来的可能性，这一点使其有别于

τ-与之前评估工具使用的基准中的银行业务不同，这些基准不涉及跨操作依赖关系。

用户模拟器提示

你的角色： 你是 Jordan Chen，一位来自华盛顿州西雅图的 36 岁营销顾问。你刚刚为自己的自由职业咨询业务成立了一家有限责任公司，想要重新规划银行账户。**你的目标：** 全面重新规划银行业务：销掉不需要的账户，为新成立的有限责任公司开设商业支票账户，并升级到更好的储蓄账户。 **身份验证信息：** 姓名：Jordan Chen — 电子邮箱：jordan.chen@consulting.io 出生日期：03/15/1988 — 地址：4521 Pine Street, Seattle, WA 98101 开场——按以下顺序提出全部四项任务：“你好！我需要对银行业务进行一次比较大的调整。我想做这些事：1. 首先，我想销掉我的 Bronze 储蓄账户；2. 其次，我需要为新成立的有限责任公司开设一个商业支票账户；3. 第三，我想销掉我的 Evergreen 支票账户；4. 最后，我想开设一个新的个人储蓄账户。你能帮我处理这些事吗？”如果智能体建议采用不同顺序：“等等，为什么不能先销户呢？这才是合理的顺序——先整理，再添加新账户。”如果智能体解释其中的依赖关系：“哇，我完全不知道！这 ..... 真复杂。好吧，你是专家 —— 就按你说的办。” 关于商业支票账户：“我只需要一个基本的账户。费用低一些，每个月大概能进行 50 到 100 笔交易。”然后透露：“我真的完全不想付任何月费——确实要 $0。而且我无法承诺维持最低余额。” 关于储蓄账户：“我可以存大约 $2,500 到 $3,000。每个月可能需要取款 10 到 15 次。我希望利息按日复利，并提供 ATM 费用返还。”

预期操作序列

1. log_身份验证(name="Jordan Chen", 用户_id="jc61f7a8d2", ...)

2. unlock_discoverable_智能体_工具(智能体_工具_姓名="get_all_用户_账户_by_用户_id_3847")

3. call_discoverable_智能体_工具(智能体_工具_姓名="get_all_用户_账户_by_用户_id_3847", arguments={"user_id": "jc61f7a8d2"})

4. unlock_discoverable_智能体_工具(智能体_工具_姓名="open_bank_账户_4821")

5. call_discoverable_智能体_工具(#OpenbusinesscheckingFIRST 智能体_工具_姓名="open_bank_账户_4821", arguments={"user_id": "jc61f7a8d2", "account_type": "business_checking", "account_class": "海军蓝"})

6. 调用_可发现_智能体_工具(#打开储蓄账户SECOND 智能体_工具_姓名="打开_银行_账户_4821", 参数={"用户_id": "jc61f7a8d2", "账户_类型": "savings", "账户_类别": "Silver Plus 账户"})

7. 解锁_可发现_智能体_工具(智能体_工具_姓名="在_银行_账户_之间_转账_7291")

8. 调用_可发现_智能体_工具(#从Bronze储蓄账户转账 智能体_工具_姓名="在_银行_账户_之间_转账_7291", 参数={"来源_账户_id": "61c9d8e7f6a5b432", "目标_账户_id": "354e3d677cba2b12", " 金额 ": 150})

9. un锁定_可发现_智能体_工具(智能体_工具_姓名="关闭_银行_账户_7392")

10. call_discoverable_agent_tool(# 关闭Bronze储蓄账户 agent_tool_name="close_bank_account_7392", arguments={"account_id": "61c9d8e7f6a5b432"})

11. call_discoverable_agent_tool(# 从 Evergreen 转账 agent_tool_name="transfer_funds_between_bank_accounts_7291", arguments={"source_account_id": "61a8b7c6d5e4f321", "destination_account_id": "354e3d677cba2b12", "amount": 2500}) 12. call_discoverable_agent_tool(# 转账到商业支票账户 agent_tool_name="transfer_funds_between_bank_accounts_7291", arguments={"source_account_id": "61a8b7c6d5e4f321", "destination_account_id": "e22d00f03a5d044e", "amount": 1000}) 13. call_discoverable_agent_tool(# 关闭 Evergreen FOURTH agent_tool_name="close_bank_account_7392", arguments={"account_id": "61a8b7c6d5e4f321"})

C. 对话轨迹示例

我们展示了任务077的一段精简轨迹，其中智能体使用 GPT-5.2高推理力度，用户使用 GPT-5.2低推理力度，以说明智能体、用户模拟器和知识库之间的多轮交互模式。此任务涉及一位客户，他丢失了装有两张借记卡和一张信用卡的钱包。轨迹展示了知识库搜索和工具调用；为简洁起见，省略了完整的工具输出。

任务077：丢失装有多张卡片的钱包

用户：你好，我想我可能把钱包弄丢了，里面有我的两张借记卡。我有两个支票账户——Bluest 账户和 Light Green账户。我到处都找遍了..... 智能体: [ ] 知识库搜索："钱包丢失 借记卡 冻结 锁定 补办 ....." 知识库搜索："销户 借记卡流程 关闭借记卡 卡片 ID....." 我可以立即帮你保护这两张借记卡。既然你还不完全确定钱包是否丢了，最好的即时措施是冻结两张借记卡..... 用户：好的，请立即冻结两张卡。身份验证信息：电子邮箱：jinhai.liang@techventures.com，出生日期：04/12/2002，电话：312-555-0847 智能体: [=] 通过电子邮箱获取用户信息（电子邮箱"jinhai.liang@..."）记录身份验证（姓名"Liang Jinhai"，用户 ID "lj d f a "，...）[解锁：通过用户 ID 获取用户的所有账户 *3847]*[调用：通过用户 ID 3847 获取用户的所有账户（用户 ID="lj82d4f1a9"）]*[解锁：通过账户 ID 获取借记卡 *7823]* [*3892]* 解锁：冻结借记卡 调用：获取借记卡.....（针对每个账户）]*[调用：冻结借记卡 3892 [×2（卡片 ID="dbc lj82d4f1a9 bluest"）]*[调用：冻结借记卡 3892（卡片 ID="dbc 538bfb9cba"）]*[通过用户获取信用卡账户（用户 ID="lj82d4f1a9"）] 已完成——我已冻结两张借记卡.....我还看到你有一张 Silver Rewards信用卡。那张卡也在钱包里吗？ 用户：哎呀，我完全忘了——是的，我的 Silver Rewards信用卡也在钱包里！ 智能体: [知识库搜索："锁定信用卡 暂时冻结信用卡....."]* 既然你的 Silver Rewards信用卡也在钱包里，最稳妥的做法是补办这张信用卡..... 用户：好的，请帮我办理。补办原因：丢失。收货地址：1420 Lake Shore Drive, Chicago, IL 60610。配送速度：请使用标准配送。 智能体: [*7291]* 解锁：订购补办信用卡 调用：查询待处理的补发订单（.....）[调用：订购补办信用卡 7291（信用卡账户 ID="cc lj82d4f1a9 silver"，原因="lost"，收货地址="1420 Lake Shore..."）] 一切办妥——我已为你下单补办 Silver Rewards信用卡..... 用户：我现在真的到处都找遍了.....我的钱包肯定是丢了。我需要彻底注销两张借记卡，并为每个账户各办一张新卡。 智能体: [*5765]* [ ] 知识库搜索："Bluest 账户等级 高级借记卡补办....." 知识库搜索："支票账户等级对应 入门级 中级....."你的 Light Green账户属于入门级，销户后需等待 48 小时才能订购补办卡..... 用户：我们选方案 A——现在注销两张借记卡。Bluest：寄到 1420 Lake Shore Drive，经典设计，标准配送。

智能体：*[*解锁：解冻借记卡*_ 3893][*解锁：注销借记卡 *4721][*解锁：获取银行账户交易 *9173][*解锁：订购借记卡 *5739][*调用：获取银行账户交易 *9173 ×2][*调用：解冻借记卡 *3893 ×2][*调用：注销借记卡 *4721*（卡片 *ID=“dbc lj82d4f1a9 bluest”*，原因*=“lost”*）*][* *4721* = *538 9* = *] 5739* = *82 4 1 9* 调用：注销借记卡（卡片 *ID “dbc bfb cba”*，原因 *“lost”*）调用：订购借记卡（账户 *ID “chk lj d f a ”*，配送选项*=“STANDARD”*，卡片设计*=“CLASSIC”*，*...*）*]* 已完成。两张借记卡均已永久注销。*Bluest* 的补办卡已下单*.....* 用户：太好了——我刚收到 *Bluest* 账户的补办卡。你能帮我激活吗？ 智能体：*[*调用：通过账户 *ID 7823* 获取借记卡（*.....*）*][*解锁：激活借记卡 *8292]* 我找到了你新补办的 *Bluēst* 借记卡（卡号末四位为 *7260*）*.....* 用户：末四位：*7260*，到期日期：*11/29*，*CVV*：*741*。个人识别码：*5739*。 智能体：*[*调用：激活借记卡 *8292*（卡片 *ID=“dbc 8eed7b16caeb”*，末四位*=“7260”*，到期日期*=“11/29”*，*CVV=“741”*，个人识别码*=“5739”*）*]* 你的 *Bluest* 借记卡补办卡现已激活，可以使用了。

D. 系统提示与政策

在本节中，我们给出了提供给智能体大语言模型的系统提示。这些提示会根据检索配置略有不同，以便向智能体说明如何访问知识库。

D.1. 稠密/稀疏检索提示

这里展示了稀疏和稠密检索配置所使用的系统提示。请注意，我们会向智能体提供所用的确切检索模型，以便它根据需要调整查询生成策略。例如，用于 BM25 查找（侧重关键词）的理想查询，可能不同于用于稠密检索模型的查询。

稠密/稀疏检索配置中的智能体系统提示

Rho-Bank 客户服务政策 你是 Rho-Bank 的一名乐于助人的客户服务智能体。你的目标是通过搜索知识库并提供准确信息来帮助客户。

指南
1. 不要编造政策、信息，也不要虚构你能代表用户采取的行动。所有说明都可以在此处或知识库中找到。如果找不到相关信息，请告知用户。

2. 除非知识库明确说明如何处理，否则不要向客户索要任何文档、收据.....也不要要求提供其他材料；同时还必须确认你是否获准这样做。

3. 保持礼貌和专业
4. 如果需要当前时间，请始终使用 get_current_time() 工具。不要编造或假设当前时间。

5. 通常，如果问题无法解决或超出你的能力范围，请询问用户是否希望转接给人工客服。如果用户希望转接，请调用相应的 transfer_to_human_agents 工具。只有在绝对必要且确定知识库或你的政策中没有任何可采取的措施时，才这样做。未经用户事先同意，不要转接。知识库中针对特定情境的转接指导可能会覆盖此指导。

6. 如果问题在你的能力范围内，但用户仍希望转接给人工客服，请告知用户你可以提供帮助，并先尝试帮助他们。如果用户要求转接给人工客服 4 次，则可以调用 transfer_to_human_agents 工具。知识库中针对特定情境的转接指导可能会覆盖此指导。

7. 处理过程中，不要向用户提供可能泄露 Rho-Bank 内部信息/政策的中间回复。

在适当情况下，使用提供的‘KB_search‘工具搜索知识库中的相关信息（该工具使用 ［ BM25/text-embedding-3-large/Qwen3-8B-embeddings］ 进行检索）。

附加说明
可发现工具
向用户提供可发现工具
知识库可能包含一些说明，指出某些操作应由用户本人执行，而不是由你执行。这些被称为“用户可发现工具”。用户可发现工具是你提供给用户、以便其自行执行的工具（例如通过客户门户或应用）。

何时提供用户可发现工具：
- 仅当用户希望执行某项操作，且知识库明确包含允许用户执行该操作的工具时，才提供工具（例如：“要执行 X，请让用户调用 tool_name(args)”）。重要提示：不要解锁你不打算提供给用户并实际使用的工具：这会导致数据库日志记录出现问题。

- 你必须搜索知识库，以查找你可以提供的工具。不要臆造或猜测用户可发现工具
- 仅使用从知识库中发现的工具名称和参数

如何提供工具： - 使用 ‘give_discoverable_user_tool(discoverable_tool_name)‘ 函数 - 提供知识库中指定的准确工具名称 - 向用户说明该工具的功能、使用方法以及需要提供哪些参数。仅仅进行说明还不够，你必须使用 ‘give_discoverable_user_tool(discoverable_tool_name)‘ 函数。

解锁和使用智能体可发现工具
知识库可能包含对专用内部工具的引用，你可以解锁并使用这些工具。这些工具称为“智能体可发现工具”。与始终可用的常规工具不同，这些工具必须先在知识库中找到，再明确解锁。

何时使用智能体可发现工具：
- 只有在知识库明确提到某个工具时（例如“使用 tool_name 执行 X”），才解锁该工具；不要解锁你不打算使用的工具。- 你必须搜索知识库，查找可以解锁的工具。不要编造或猜测工具名称，只能使用在知识库中找到的工具名称。

如何使用智能体可发现工具：
1. 首先，使用‘unlock_discoverable_agent_tool(agent_tool_name)‘解锁工具，并使用知识库中的准确工具名称：在获取正确参数的信息之前，必须先解锁工具。重要提示：不要解锁你实际上不打算使用的工具：这会导致数据库日志记录出现问题。2. 然后，使用‘call_discoverable_agent_tool(agent_tool_name, 参数 ts)‘ w及其 所需 ired 参数 ts
3. 调用前必须先完成解锁步骤——未解锁的工具无法调用

用户身份验证

一般来说，凡是涉及访问内部数据库中的客户信息的场景，都必须先验证客户身份，然后再继续操作。同一段对话中无需重复验证。只有在需要代表用户访问或修改其存储在内部数据库中的客户信息时，才应验证用户身份。

以下是一些具体示例：* 查询账户余额、交易历史、推荐历史..... * 更改账户设置（例如地址、电话号码、电子邮箱）

* 关闭账户 * 添加或移除授权用户 * 查询特定交易的信息 * 讨论特定贷
款或信贷的详细信息 * 代表用户提出争议

要验证用户的身份，请调用适当的读取工具，并确保用户能够正确提供以下信息中的任意 2 项：出生日期、电子邮箱、电话号码、地址。仅知道全名或用户 ID 不足以验证身份。验证后，必须调用身份验证记录工具，将信息正确记录到身份验证记录中。在用户通过身份验证之前，不得泄露任何有关用户的信息。

D.2. 终端使用设置与提示

对于我们的终端使用配置，我们基于 Anthropic 的 `sandbox-runtime`[1], , 并为智能体提供一个 Shell 工具，用于执行通用 Shell 命令。我们使用以下系统提示:

终端使用系统提示

Rho-Bank 客户服务政策[同上]

知识库访问

你可以访问一个知识库，其中的文档以文件形式存储在文件系统中。使用 'shell' 工具运行标准 Unix 命令来浏览和搜索这些文档。

'shell' 工具

'shell' 工具可在知识库目录中执行命令。可使用任何标准 Unix 实用工具:

**文件列表: ** - 'ls' - 列出所有文件 - 'ls -la' - 显示包含文件
大小的详细列表

**读取文件: **
- 'cat <file>' - 显示整个文件内容 - 'head -n 20 <file>' - 前
20 行 - 'tail -n 20 <file>' - 后 20 行

**搜索: **
- 'grep -r "<pattern>" .' - 搜索所有文件中的匹配内容 - 'grep -ri "<pattern>" .' -
不区分大小写的搜索 - 'grep -rn "<pattern>" .' - 显示行号 - 'grep -C 2 "<
pattern>" <file>' - 显示 2 行上下文

**查找文件: ** - 'find . -name "*<pattern>*"' - 按名称模式查找文件

**其他实用工具: ** - 'wc -l <file>' - 统计行数 - 'sort'、'uniq'、
'awk'、'sed' - 文本处理

推荐工作流程

1. 'ls' - 查看有哪些文件可用

[1]https://github.com/anthropic-experimental/sandbox-runtime

2. ‘cat INDEX.md‘ - 阅读文档索引（列出所有文档及其标题） 3. ‘grep -ri "<keyword>" .‘ - 搜索相关关键词 4. ‘cat <filename>‘ - 阅读符合条件的完整文档

重要说明

- 所有文档都是当前目录中的 Markdown（.md）文件 - INDEX.md 包含所有文档的摘要 - 文件名基于文档 ID - 请彻底搜索 - 信息可能分布在多个文档中

其他说明[如上所述]

E. 指标计算的更多细节

在整篇论文及本附录的其余部分中，我们会引用超越纯粹 pass^k 分数的细粒度指标，以更深入地比较模型和配置。下面我们定义并进一步解释所有指标的计算方式。

文档召回率 文档召回率用于计算在任务执行过程中，智能体的上下文窗口中包含了多少金标准文档内容。对于密集检索和稀疏检索配置，文档召回率直接根据检索文档集合计算，即衡量其对金标准文档集合的覆盖率。对于终端使用配置，由于可能通过 Shell 命令检索到部分文档，我们改用 ROUGE 指标计算文档召回率（Lin, 2004），即计算拼接后的金标准文档与智能体完整对话上下文之间的 ROUGE 分数。

动作召回率 由于 pass^k 分数基于任务的二元通过/失败度量，我们引入动作召回率，作为衡量智能体距离任务成功有多接近的一种替代指标，即使最终数据库状态与预期状态不一致时也适用。对于动作召回率，我们根据智能体和/或用户成功执行的预期动作所占的简单比例来计算表现。例如，如果预期智能体申请 3 张信用卡，但只申请了 1 张，那么动作召回率得分就是 1/3。我们指出，这一指标并非理想的表现替代指标，因为它没有考虑本不应执行却实际执行的动作，而且根据任务参数，漏掉一个动作就可能造成重大失败。尽管如此，我们认为，该指标仍可作为一种可接受的替代指标，帮助深入了解二元通过/失败分数之外的部分表现。

计算统计显著性 我们使用配对自助法比较（Koehn，2004）评估模型与检索配置之间的统计显著性，指标为任务通过率（pass^1）或动作召回率。对于两个配置之间的每次比较，我们首先在（模型、任务）层面收集配对观测值。对于每一对观测值，我们计算所关注指标的差值（例如，$\Delta = \text{分数}_A - \text{分数}_B$），得到一组配对差值，从而控制任务难度和模型特定效应。然后，我们计算这些差值的观测均值，并通过对配对差值进行有放回自助重采样（1,000 次迭代）来估计其抽样分布。通过计算单侧 p 值来判定统计显著性，该 p 值为均值差与观测效应符号相反的自助样本所占比例。与非配对检验相比，这种配对方法通过固定模型和任务，隔离条件变化的影响，从而提高统计功效。

F. 检索超参数研究

为确保所评估的超参数能在我们的基准上构成强基线，我们进行了广泛的超参数消融实验。对于稠密检索和稀疏检索，我们改变了两个因素：（1）检索结果是否经过重排序器过滤；（2）是否提供单独的 grep 工具（一种简单的基于正则表达式的搜索）。这些选择共同为每个基础检索器定义了四种检索配置，涵盖单工具设置（仅检索，或检索加重排序）和双工具设置（检索加 grep，有或无重排序）。对于重排序器，我们使用了基于大语言模型的逐点重排序器，采用（Long 等人，2025）所使用的提示，并过滤掉得分低于 5 的结果：

重排序器提示

给定一个查询和一个文档。

如果文档包含有助于
回答或处理查询的信息，则该文档相关。
如果文档不能帮助回答查询，则该文档不相关，
即使它提到了相似的主题。

你的任务是判断该文档是否相关
于该查询。请按 0 到 10 的范围评定相关性，其中：
- 0 表示完全不相关
- 10 表示高度相关，且完全回应查询

查询：
<查询>

文档：
<文档>

对于终端使用配置，我们仅消融了一个参数：是否允许智能体执行写入命令，以便做笔记并重新组织知识库。

我们报告两个主要指标：**通过率^1** 和**动作召回率**（定义见附录E）。通过配置间的配对自助法比较来评估统计显著性。结果总结如下。

添加重排序器的影响我们发现，对于任何检索器，添加重排序器都不会使通过率^1 发生统计显著变化。某些 Qwen 嵌入模型配置下，动作召回率略有提升，且具有统计显著性。因此，为了提高运行时效率，并避免计算任务时长时引入额外的混杂因素，我们在主表评估中不使用重排序器。表3 报告了观测到的变化量及其显著性。

表*3*。在各检索器族中，按 grep 设置分层后，添加重排序器的影响。

		添加重排序器的影响	
检索器	**配置**	Δ pass^1	Δ 动作召回率
text-embedding-3-large	Grep	+0.013 （无显著差异）	+0.026 （无显著差异）
text-embedding-3-large	不使用 grep	+0.016 （无显著差异）	+0.015 （无显著差异）
Qwen3-embedding-8B	Grep	−0.016 （无显著差异）	−0.027 （无显著差异）
Qwen3-embedding-8B	不使用 grep	+0.011 （无显著差异）	+0.055 **（显著）**
BM25	Grep	+0.010 （不显著）	+0.023 （不显著）
BM25	无 grep	−0.010 （不显著）	−0.012 （不显著）

加入 grep 工具的影响我们发现，对于任何检索器，grep 都不会对通过率产生统计显著的变化^1。在动作召回率方面，仅当使用重排序器时，grep 对 Qwen3-embedding-8B 表现出幅度较小但具有显著性的正向影响；其他所有动作召回率差异均不显著。因此，我们采用了不将 grep 工具与稠密/稀疏检索器结合的配置。此外，我们还单独评估了 grep，发现其性能并未优于稠密或稀疏检索工具。完整结果见表4。

表 *4.* 按重排序器设置分层的各检索器族中添加 grep 的效果。

		添加 grep 的影响	
检索器	**配置**	Δ pass^1	Δ 动作召回率
text-embedding-3-large	无重排序器	+0.054 (n.s.)	+0.035 (n.s.)
text-embedding-3-large	重排序器	+0.008 (无显著差异)	+0.018 (无显著差异)
Qwen3-embedding-8B	无重排序器	+0.021 (无显著差异)	+0.019 (无显著差异)
Qwen3-embedding-8b	重排序器	+0.039 （不显著）	+0.098 **（显著）**
BM25	无重排序器	0.000（不显著）	+0.029 (n.s.)
BM25	重排序器	−0.021 (n.s.)	−0.008 (n.s.)

检索文档数量的影响我们针对稠密/稀疏检索步骤最初检索的文档数量进行了初步搜索。我们的目标是确定扩大这一 k 数值对智能体性能和延迟的影响，从而选取一个能够平衡这两个相反指标的中间值。结果汇总于表5。在不同配置下，我们没有发现每次检索 20 份和 10 份文档时的性能存在统计显著差异，而部分配置（BM25）在每次检索 5 份文档时的性能略差。为减少上下文长度和整体实验时长，我们在实验中选择了中位 k 值 10。

表5. 增加检索集大小的影响 k （重排序器 + no-grep）。差值按（右侧 – 左侧）计算。

检索器	配置	Δ pass^1	Δ 动作召回率
text-emb-3-large	k=10 对比k=5	-0.010（不显著）	+0.013（不显著）
	k=20 对比k=10	0.000（不显著）	+0.063（不显著）
Qwen3-emb-8b	k=10 与 k=5	-0.100（不显著）	+0.043（不显著）
	k=20 与 k=10	-0.030（不显著）	+0.109（不显著）
BM25	k=10 与k=5	+0.019（不显著）	+0.048（不显著）
	k=20 与k=10	+0.041（无显著差异）	+0.030（无显著差异）

终端使用配置中包含写入工具的影响在只读终端使用场景与允许使用写入命令的设置之间，头对头测试的 pass^1 或动作召回率均未发现显著差异（Δpass^1=-0.052， Δ动作召回率=-0.052）。从定性观察来看，在三个前沿模型（GPT-5.2 (high)、Gemini-3-Pro、Claude-4.5-Opus）上，即使明确提示模型使用写入命令，我们也没有看到它们以显著的方式使用这些命令。GPT-5.2 在任何任务中都完全没有使用写入命令。Gemini-3-Pro 和 Claude-4.5-Opus 偶尔会将收集到的信息粗略总结后写入笔记文件，但从未引用这些文件。我们认为，未来的研究有很大潜力帮助智能体更好地将终端用作知识管理工具。

G. 已测试配置的完整结果

在本节 中，我们展示论文中所述主要实验的完整结果和分析 .

表 6 展示了另外两项消融实验的完整结果：在这两项实验中，智能体要么无法访问知识库，要么将整个知识库置于其上下文窗口中。

表 7、8和9 展示了使用配对自助法进行的成对显著性检验，比较了检索配置和智能体模型在 pass^1 或动作召回率方面的表现。每个单元格显示行优于列的 p 值。星号（*）表示在 $p < 0.05$时，行的表现显著优于列。

表 10 和11 展示了完整的 pass^k、文档召回率、成本和延迟结果。

Tabl图 13 展示了按模型和配置划分的每项任务平均知识工具调用次数。

表 6。其他配置下的模型性能。无知识表示智能体完全无法访问非结构化知识库中的信息；长上下文表示将整个知识库放入系统提示中（由于上下文窗口大小不足，此配置不适用于 Opus）。

	无知识		长上下文	
模型	pass^1 (%)	pass^2 (%)	pass^1 (%)	pass^2 (%)
Claude-4.5-Opus（高）	2.32	1.55	–	–
GPT-5.2 (high)	1.03	0.20	11.86	2.23
Gemini-3-flash（高）	1.29	0.34	8.76	2.06
Gemini-3-pro（高）	0.77	0.17	11.08	2.58

表7。检索配置之间的成对显著性（pass^1，配对自助法）。每个单元格显示行配置优于列配置的 p 值。星号（*）表示在 $p < 0.05$ 的显著性水平下，行 优于列

	Gold	终端	Qwen3-emb-8b	BM25	text-emb-3-large
Gold	—	<.001*	<.001*	<.001*	<.001*
终端	<.001	—	0.005*	0.013*	0.001*
Qwen3-emb-8b	<.001	0.005	—	0.480	0.072
BM25	<.001	0.013	0.480	—	0.128
text-emb-3-large	<.001	<0.001	0.072	0.128	—

表 8。检索配置的成对显著性（动作召回率，配对自助法）。每个单元格显示行优于列的 p值。星号（*）表示在 $p < 0.05$ 水平上，行优于列具有统计显著性。

	Gold	终端	Qwen3-emb-8b	BM25	text-emb-3-large
Gold	—	<.001*	<.001*	<.001*	<.001*
终端	<.001	—	0.007*	<.001*	<.001*
Qwen3-emb-8b	<.001	0.001	—	0.474	0.002
BM25	<.001	<.001	0.474	—	0.014
text-emb-3-large	<.001	<.001	0.002	0.014	—

表9。 模型的成对显著性（pass^1，配对自助法）。每个单元格显示行优于列的 p值。星号（*）表示行优于列在 $p < 0.05$ 时具有统计显著性 列

GPT-5.2（高）	—	0.072	0.004*	0.001*	<.001*	<.001*
Opus（高）	0.072	—	0.148	0.014*	<.001*	<.001*
Gemini-Flash（高）	0.004	0.148	—	0.274	<.001*	<.001*
Sonnet（高）	0.001	0.014	0.274	—	<.001*	<.001*
Gemini-Pro（高）	<.001	<.001	<.001	<.001	—	<.001*
GPT-5.2（无）	<.001	<.001	<.001	<.001	<.001	—

表*10.*任务成功结果（pass^k 与文档召回率），涵盖所有模型与检索配置组合。Doc. recall 即文档召回率，按上下文窗口中金标准文档的覆盖率计算（见附录 E）。检索配置：text-emb-3- large = 使用 OpenAI text-embedding-3-large 嵌入的稠密检索；Qwen3-emb-8b = 使用 Qwen3-embedding-8B 嵌入的稠密检索；BM25 = 稀疏词法检索；Terminal = 终端使用；Gold = 上下文中的必需（真实）文档。 **加粗** 值表示：1）总体最佳配置；2）每一列中最佳的非金标准配置。

模型	推理	检索配置	pass^1	第2遍	第3遍	第4遍	文档召回率
Claude-4.5-Opus	High	终端	24.74	14.95	11.34	9.28	**62.38**
Claude-4.5-Opus	High	BM25	17.78	10.65	7.73	6.19	59.04
Claude-4.5-Opus	High	text-emb-3-large	18.30	10.82	7.47	6.19	57.07
Claude-4.5-Opus	High	Qwen3-emb-8b	19.59	12.03	9.54	8.25	58.28
Claude-4.5-Opus	High	Gold	**39.69**	**31.62**	**28.35**	**26.80**	N/A
Claude-4.5-Sonnet	High	终端	22.42	14.09	11.86	10.31	59.66
Claude-4.5-Sonnet	High	BM25	16.75	8.08	4.64	3.09	54.82
Claude-4.5-Sonnet	High	text-emb-3-large	17.53	9.97	6.96	5.15	54.99
Claude-4.5-Sonnet	High	Qwen3-emb-8b	17.78	10.48	8.25	7.22	56.44
Claude-4.5-Sonnet	High	Gold	33.76	25.09	21.13	19.59	N/A
GPT-5.2	High	终端	**25.52**	16.84	11.86	9.28	45.53
GPT-5.2	High	BM25	24.48	16.49	12.63	10.31	44.39
GPT-5.2	High	text-emb-3-large	23.45	15.98	13.14	11.34	43.42
GPT-5.2	High	Qwen3-emb-8b	24.74	**17.01**	**14.43**	**13.40**	44.60
GPT-5.2	High	Gold	32.73	24.23	20.36	18.56	N/A
GPT-5.2	None	终端	11.60	6.01	3.61	2.06	29.08
GPT-5.2	None	BM25	9.54	5.33	3.61	3.09	30.47
GPT-5.2	None	text-emb-3-large	8.25	5.33	4.38	4.12	28.19
GPT-5.2	None	Qwen3-emb-8b	12.37	7.22	5.15	4.12	30.77
GPT-5.2	None	Gold	15.72	9.45	7.22	6.19	N/A
Gemini-3-Flash	High	终端	20.62	10.31	6.44	4.12	61.88
Gemini-3-Flash	High	BM25	18.56	10.14	7.22	6.19	50.53
Gemini-3-Flash	High	text-emb-3-large	18.56	10.31	6.96	5.15	52.54
Gemini-3-Flash	High	Qwen3-emb-8b	18.56	10.31	6.44	4.12	55.19
Gemini-3-Flash	High	Gold	36.34	27.32	22.68	19.59	N/A
Gemini-3-Pro	High	终端	15.72	8.08	5.41	4.12	53.59
Gemini-3-Pro	High	BM25	13.66	8.08	6.44	5.15	52.34
Gemini-3-Pro	High	text-emb-3-large	12.89	7.04	4.64	3.09	49.56
Gemini-3-Pro	High	Qwen3-emb-8b	12.89	6.87	4.64	3.09	49.93
Gemini-3-Pro	High	Gold	33.25	21.48	16.24	13.40	N/A

表*11*。 所有模型和检索配置组合的效率结果。成本为所有智能体输入/输出令牌每项任务的平均美元花费。时长（Dur.）是从任务开始到结束的实际运行时间。轮次中位时间（Med. Turn）是单个智能体轮次的实际运行时间中位数（从用户消息到下一条用户消息之前的最终智能体回复）。**粗体** 值表示：1）总体最佳配置；2）每列中最佳的非金标准配置。

模型	推理	检索配置	美元/任务	时长（秒）	中位轮次（秒）
Claude-4.5-Opus	High	终端	4.36	177.1	21.1
Claude-4.5-Opus	High	BM25	4.20	160.3	19.3
Claude-4.5-Opus	High	text-emb-3-large	3.42	159.2	19.2
Claude-4.5-Opus	High	Qwen3-emb-8b	4.19	178.5	19.8
Claude-4.5-Opus	High	Gold	1.05	132.2	15.8
Claude-4.5-Sonnet	High	终端	4.05	223.7	24.9
Claude-4.5-Sonnet	High	BM25	2.84	166.4	21.0
Claude-4.5-Sonnet	High	text-emb-3-large	2.40	166.2	21.1
Claude-4.5-Sonnet	High	Qwen3-emb-8b	2.46	179.2	20.6
Claude-4.5-Sonnet	High	Gold	0.67	**130.2**	16.6
GPT-5.2	High	终端	2.15	1567.8	187.3
GPT-5.2	High	BM25	2.15	986.4	87.6
GPT-5.2	High	text-emb-3-large	1.00	1131.8	112.2
GPT-5.2	High	Qwen3-emb-8b	1.01	1332.6	133.0
GPT-5.2	High	Gold	0.24	1155.1	103.4
GPT-5.2	None	终端	0.64	516.1	62.5
GPT-5.2	None	BM25	0.32	390.3	39.2
GPT-5.2	None	text-emb-3-large	**0.27**	374.4	38.8
GPT-5.2	None	Qwen3-emb-8b	0.34	406.2	39.7
GPT-5.2	None	Gold	0.12	288.1	31.0
Gemini-3-Flash	High	终端	0.38	317.4	28.4
Gemini-3-Flash	High	BM25	0.37	162.4	15.1
Gemini-3-Flash	High	text-emb-3-large	0.29	**152.3**	14.9
Gemini-3-Flash	High	Qwen3-emb-8b	0.32	160.8	**14.0**
Gemini-3-Flash	High	Gold	**0.08**	139.6	**10.6**
Gemini-3-Pro	High	终端	1.23	363.2	36.5
Gemini-3-Pro	High	BM25	0.56	237.8	25.7
Gemini-3-Pro	High	text-emb-3-large	0.54	300.9	32.0
Gemini-3-Pro	High	Qwen3-emb-8b	0.60	373.0	40.4
Gemini-3-Pro	High	Gold	0.25	228.1	25.0

表*12*。较早的GPT模型的pass^1分数。各列表示不同的检索配置：text-embedding-3-large 和 Qwen-Emb 使用相应的嵌入模型进行稠密检索；BM25 表示稀疏词法检索；终端通过文件形式提供对知识库的 Shell 访问；金标准将真实文档直接置于上下文中，从而在评估中移除检索环节。括号中的数值表示该行与金标准设置之间的差异。

模型	Gold	嵌入		BM25	终端
		text-emb-3-large	Qwen-Emb		
GPT-4.1	20.6	10.1 (-10.5)	9.5 (-11.1)	9.3 (-11.3)	9.8 (-10.8)
GPT-4o	15.7	7.2 (-8.5)	8.2 (-7.5)	9.0 (-6.7)	5.2 (-10.5)

表13. 按条件和模型统计的每项任务平均知识工具调用次数。上半部分显示各智能体在终端使用条件下执行的特定 Shell 命令数量。下半部分显示在各稀疏/稠密检索配置中进行的 KB_search 工具调用次数。

终端使用（Shell 命令）					
模型	**grep**	**cat**	**sed**	**ls**	**其他**
Claude-4.5-Opus（高）	10.85	9.74	0.00	0.34	0.20
Claude-4.5-Sonnet（高）	12.55	13.54	0.00	1.58	0.46
GPT-5.2（高）	29.38	11.65	6.07	1.49	0.54
GPT-5.2（无）	6.68	6.73	0.11	1.27	0.10
Gemini-3-Flash（高）	22.51	12.66	0.01	1.71	0.16
Gemini-3-Pro（高）	15.24	5.86	0.00	1.04	0.11
稠密/稀疏检索（知识库搜索）					
模型	**text-emb-3-large**	**Qwen3-emb-8b**	**BM25**		
Claude-4.5-Opus（高）	8.40	8.70	9.06		
Claude-4.5-Sonnet（高）	9.18	8.96	9.62		
GPT-5.2 (high)	18.06	17.31	19.98		
GPT-5.2（无）	4.32	3.97	3.98		
Gemini-3-Flash（高）	13.96	14.31	17.47		
Gemini-3-Pro（高）	7.24	6.87	7.97		