

# 语义图建模：智能体构建可执行知识图谱支持长程工作

**Yanming Guo**

悉尼大学

AIDC

yguo0337@uni.sydney.edu.au

**Haixin Wang**

AIDC

nix@ai-dc.ai

**YanJun Lu**

AIDC

lucas@ai-dc.ai

## 摘要

在组织内长期工作的语言模型智能体，反复从原始表、制度文档和自身历史中推导相同知识。我们研究语义图建模：工作开始前，智能体将领域的数据来源和制度编译为可执行知识图谱，包括映射到记录系统的带类型对象与链接、以条件编码制度的动作、函数以及在变更时触发的自动化规则，随后通过带类型接口开展工作。我们形式化定义图及建模算子，说明此类图能够及不能节省什么，并提出 OpsWEEK，一个模拟企业工作周的生成器，包含传统 ERP、制度文本、事件及四十个按状态判分的请求。使用 DeepSeek V4.1 Flash，智能体在 7.3 分钟内构建了初始图，成本为 0.82 CNY。在六个工作周中，使用图的请求成功率为 99.5%，同一智能体通过 SQL 操作原始 ERP 时为 97.7%；在这一接近上限的水平上，差异无定论。长上下文提示 (77.8%) 和检索 (24.1%) 这两个单次调用下限基线远远落后。使用图时，每请求 token 数为原始智能体的 1.66 $\times$ ，部分原因是每次调用都重新发送建模对话记录 (清空对话后其 token 数减半，差距缩小到 1.27 $\times$ )，且未收回准备成本。图改变了工作方式：自动化规则无需模型调用便满足了 78.6% 的持续指令义务，每次写入都通过由引擎检查的动作。然而，原始智能体也将制度编译成了自己的辅助脚本库；图增加的是强制执行与触发器，而非编译后的知识。

## 1 引言

语言模型智能体正在从回答请求转向持续开展工作：它们运行订单服务台、监控库存或核对账目，持续数日，经历数百次模型调用 (Xu et al., 2025a; Backlund & Petersson, 2025)。这类工作的难点很少在于某一个步骤，而在于每一步所需的知识，包括列的含义、两张表的连接方式、写入适用的制度及已处理的事件，分散在原始表、制度文档和智能体自身历史中，智能体反复从中重新推导。长上下文的表现会退化 (Liu et al., 2024a; Hsieh et al., 2024)，检索返回片段 (Lewis et al., 2020)，有数据库访问权限的智能体则在每个请求中重新计算同样的连接。在为三家公司约三百个智能体提供服务的生产部署中，我们观察到这三种现象<sup>1</sup>：智能体为回答反复出现的运营问题而重读原始 ERP 导出数据；一家公司的智能体运行了约 1,800 个重复检查数据的定时任务，因成本问题被关闭；长时间的运营轮次则在中途超时。

结构化知识是经典的解决方法。知识图谱、基于本体的数据访问和语义层赋予数据明确的类型与语义，由引擎而非模型计算答案 (Poggi et al., 2008; Hogan et al., 2021)；语言模型通过本体回答企业问题，比通过原始模式更准确 (Sequeda et al., 2024)。但对于长程智能体，这些模型存在三方面不足。它们是读模型：约束写入的规则仍留在提示中。它们是被动的：数据变化时不会自动响应，因此智能体需要轮询。它们还由其他人或系统构建，即人工或离线抽取流水线 (Zhang & Soh, 2024)，因而构建成本从未与使用时的节省相权衡。商业数据平台提供运营本体，将对象类型绑定到数据源，并配以带类型的动作和自动化规则。在我们筛选的 475 篇候选论文中 (附录 H)，最接近的工作让智能体构建只读本体层，并按单个问题评估 (Chong et al., 2026)；我们未发现受控研究考察可执行模型能否帮助语言模型智能体开展长时间连续工作，或智能体能否独立构建这样的模型。

我们研究语义图建模 (SGM)：工作开始前，智能体将领域的数据来源和制度编译为可执行语义图，包括映射到数据源的带类型对象与链接、以提交条件编码制度并将效果写回记录系统的动作、计算派生量的函数，以及数据变化时触发的自动化规则；随后通过带类型的命令行

<sup>1</sup>作者于 2026 年 9 月和 10 月在自身部署中观察所得；计数为近似值。

接口开展工作。三方面可能因此改变：上下文可保存数据视图而非数据本身；由引擎而非模型检查写入是否符合制度；持续指令转化为触发器，而非轮询。我们提出三个问题。**RQ1**：在演化数据上的长时间连续工作中，相比长上下文提示、检索增强生成（RAG）及操作原始数据源的工具型智能体，语义图能否使智能体更准确、更便宜、更快？**RQ2**：智能体能否独立构建图，自建图与专家图相比如何，建模何时能够回本？**RQ3**：图的哪些部分产生作用，是知识图谱部分（带类型对象、链接、引擎端查询），还是可执行部分（动作、函数、自动化规则）？

为回答这些问题，我们构建了 OpsWEEK，一个模拟企业运营工作周的生成器：包含传统 ERP、制度文本、事件流及每周四十个请求，由按状态判分的参考判定器评分（第 5 节）。使用 DeepSeek V4.1 Flash，得到的答案与知识图谱视角的预期不同。**RQ1**：图未使准确率产生可检测的变化，因为通过 SQL 操作原始 ERP 的智能体已经在 97.7% 的请求上成功（使用自建图时为 99.5%）；图也未节省 token：SQL 已能返回紧凑视图，构建图的对话记录随后每次调用重新发送（每请求 token 数为原始智能体的  $1.66\times$ ）。**RQ2**：智能体在 7.3 分钟内独立构建了初始图，成本为 0.82 CNY，但原始智能体也为相同制度构建了自己的辅助脚本库，图未收回准备成本。**RQ3**：阶梯中没有任何一级使准确率产生可检测的变化；可执行部分改变的是工作方式：写入通过由引擎检查的动作完成，自动化规则无需模型调用便满足了 78.6% 的持续指令义务。长上下文提示（成功率 77.8%，每请求 1.003 CNY）和检索（24.1%）远远落后。

贡献。

- 形式化定义语义图建模：由智能体构建的可执行知识图谱  $G = (\Sigma, \mu, D, \Delta)$ ，配以建模算子与带类型接口，并提出四个命题，说明此类图能够及不能节省什么（第 3 节）。
- OpsWEEK，一个长程企业运营基准生成器，包含演化数据、受制度约束的写入及持续指令，按状态判分，并提供三种数据规模（第 5 节）。
- 一个在共享生产运行时池上运行智能体的系统，隔离每次执行，通过带类型的命令行接口和事务发件箱连接工业本体内核（第 4 节）。
- 一项包含七个条件、消融阶梯、主要条件各六个会话及完整成本核算的研究。研究发现，语义图改变的是智能体的写入和响应方式，而非工作的准确率或成本；原始智能体也为相同制度编译了没有强制执行机制的辅助脚本（第 6 节）。

## 2 相关工作

附录 H 给出本节所依据的完整综述。

面向语言模型的知识图谱与语义层。知识图谱赋予实体和关系明确的类型与语义 (Hogan et al., 2021; Pan et al., 2024)。语言模型沿图路径推理 (Sun et al., 2024)，或在文本构建的图索引上检索 (Edge et al., 2024; Gutiérrez et al., 2024)。针对关系数据，基于本体的数据访问用领域概念表达查询 (Poggi et al., 2008; Calvanese et al., 2016)，工业语义层则为所有使用方一次性定义度量与连接 (dbt Labs, 2026; Google, 2026; Cube, 2026)；文本到 SQL 方法在语义依赖业务知识的企业模式上仍表现较弱 (Lei et al., 2025; Chen et al., 2024)。查询企业数据库的知识图谱而非其模式，使 GPT-4 的准确率从 16% 提升到 54%（作者自报；Sequeda et al., 2024）；语义层也被报告有类似提升 (Rumiantsau & Fokeev, 2026)。ChatDB 将数据库用作智能体的符号记忆 (Hu et al., 2023)，EvoOntology 则让构建智能体为数据智能体构建并改进本体层 (Chong et al., 2026)。这些工作都是按单个问题评估的读模型。我们加入写入路径和触发器，由执行工作的智能体构建它们，并在长时间连续会话中评估，完整核算成本。

用语言模型构建和评估本体。语言模型从文本中抽取实体、关系和模式 (Zhang & Soh, 2024; Bai et al., 2026)，并辅助构建本体 (Lo et al., 2024; Zhang et al., 2025a)；图谱依据参考图谱评分 (Mihindukulasooriya et al., 2023)，而本体工程通过能力问题和建模陷阱目录评估模型，不依赖单一的标准本体 (Grüniger & Fox, 1995; Poveda-Villalón et al., 2014)。本文中，智能体将结构化数据源与制度文本一同建模，模型除类别与关系外还包含操作和规则，并以其支持的工作及构建成本来评价。

上下文、记忆与长程智能体。长上下文的表现会随长度增加而退化 (Liu et al., 2024a; Hsieh et al., 2024; Hong et al., 2025)，这推动了检索 (Lewis et al., 2020; Li et al., 2024)。智能体记忆系

统对智能体自身的经验进行分页、摘要或结构化组织 (Packer et al., 2023; Zhang et al., 2025c), 也将其组织为时序图 (Rasmussen et al., 2025); 对于精确回忆, 由智能体控制的原始日志搜索可达到结构化记忆的水平 (Li et al., 2026)。企业智能体基准测试数据库和应用中的工具使用 (Styles et al., 2024; Yao et al., 2025; Xu et al., 2025a; Huang et al., 2026), 长时间模拟运行会偏离目标 (Backlund & Petersson, 2025); 很少有工作在报告准确率的同时报告成本 (Kapoor et al., 2024)。语义图对领域的状态和规则建模, 而非智能体的经验; OPSWEEK 加入了演化数据上的连续会话及持续指令。

守卫、编译后的知识与自动化。智能体通过工具和代码采取行动 (Yao et al., 2023; Wang et al., 2024b), 为求解器编写世界模型 (Guan et al., 2023; Tang et al., 2024), 并积累可复用技能 (Wang et al., 2024a), 也将其纳入受治理的注册表 (Guo, 2026d)。制度可编译为守卫, 在运行时检查工具调用 (Zwerdling et al., 2025; Wang et al., 2026a; Chen et al., 2025; Xiang et al., 2025; Rebedea et al., 2023; Moslemi et al., 2026)。具有守卫-阶段-里程碑生命周期的业务工件, 即变化受守卫约束并由事件驱动的以数据为中心的对象, 是最接近的形式化先例 (Nigam & Caswell, 2003; Hull et al., 2011); 事件-条件-动作规则源于主动数据库 (Widom & Ceri, 1995), 而通过模型写回数据源属于视图更新问题 (Bancilhon & Spyrtos, 1981; De Giacomo et al., 2017), 实践中通过将写入路径与读模型分离来解决 (Fowler, 2011)。语义图将这些部分结合到一个模型中, 由智能体自身通过相同的分支-验证-发布周期定义、查询和修改。

### 3 语义图建模

#### 3.1 问题设定：记录系统上的长程工作

智能体在一个领域中开展长时间会话。该领域具有一个记录系统, 其中包含关系数据源  $R = \{R_1, \dots, R_m\}$ 、书面文档  $P$  (数据字典与制度), 以及在请求之间改变  $R$  的外部事件流 (订单到达、货物发运、发票付款)。工作以带日期的请求序列  $x_1, \dots, x_n$  到达。请求可以要求获取信息 (查询), 要求在遵守制度的前提下改变  $R$  (写入), 或给出适用于未来事件的持续指令 (规则, 例如“从现在起, 每笔新 EDI 订单到达时都要检查”)。我们遵循 Guo (2026a): 智能体是作用于模型  $L$  和工具  $T$  的运行框架 (harness)  $\mathcal{H}_{L,T}$ ; 每个请求对应一次执行, 读取智能体记录  $d = (A, C || s)$ , 其中  $A$  为定义、 $C$  为上下文、 $s$  为工作空间, 然后提交新记录。会话的衡量指标包括正确完成的请求比例、及时处理的规则触发事件比例, 以及成本: 每个请求的模型调用次数、输入与输出 token 数及实际耗时。

若没有进一步的结构, 智能体学到的所有领域知识, 包括列的含义、两张表的连接方式、写入适用的规则以及已处理的事件, 都存于  $C$  或  $s$ , 每次需要时由模型重新读取。语义图建模将这些知识转移到外部、带类型且可执行的模型中, 由引擎而非模型进行求值。

#### 3.2 语义图

定义 1 (语义图). 数据源  $R$  上的语义图为  $G = (\Sigma, \mu, D, \Lambda)$ , 其中包含

- 模式  $\Sigma = (O, E, \mathcal{A}, F, \Gamma)$ : 对象类型  $O$ , 每种类型具有带类型和描述的属性及一个主键; 链接类型  $E \subseteq O \times O$ ; 动作类型  $\mathcal{A}$ , 每个动作为三元组  $(\Theta_a, c_a, \rho_a)$ , 分别表示带类型的参数、提交条件  $c_a(\theta, D)$  (接受或拒绝并给出理由), 以及向记录系统发出请求的效果  $\rho_a(\theta, D)$  (在我们的实现中,  $c_a$  和  $\rho_a$  合为一条函数规则, 通过抛出错误来拒绝); 函数  $F$ , 即作用于对象集的带类型、无副作用程序; 自动化规则  $\Gamma$ , 每条规则为二元组  $(\kappa_\gamma, \epsilon_\gamma)$ , 分别表示作用于对象集或变更集的条件, 以及应用动作、调用函数或发送通知的效果;
- 映射  $\mu$ , 将每种对象类型关联到源表或只读查询, 并将每个属性关联到列或表达式, 与基于本体的数据访问相同 (Poggi et al., 2008); 派生量由此类表达式或  $F$  中的函数给出;
- 数据  $D = \mu(R)$ , 即实例化  $\Sigma$  的带类型属性图; 每次动作和每批外部变更之后, 均从  $R$  重新同步; 若智能体也能直接写入  $R$ , 则还在每次读取前同步, 使每次读取都能看到  $R$  的当前状态;
- 日志  $\Lambda$ , 记录每次已应用动作的参数、执行者和效果。

元组  $(O, E, \mu, D)$  是通常意义上的知识图谱 (Hogan et al., 2021): 具有语义和来源的带类型实体与关系。知识图谱、语义层及基于本体的数据访问主要用作读模型。后半部分  $(\mathcal{A}, F, \Gamma)$  使图具有可执行性: 它规定领域可以如何变化及其条件、如何计算派生量, 以及数据变化时必须发生什么。商业数据平台中的运营本体具有相同结构; 我们研究它作为语言模型智能体工作基底的作用。

接口。智能体仅通过带类型的命令行接口访问  $G$ 。查询  $q \in Q$  (带过滤条件的搜索、沿链接遍历、聚合、函数调用) 返回视图  $v = q(D)$ , 只有视图进入上下文。应用  $\text{apply}(a, \theta)$  当且仅当  $c_a(\theta, D)$  成立时提交; 其效果  $\rho_a(\theta, D)$  由记录系统执行, 随后从  $R$  重新同步  $D$ 。自动化规则在引擎内部运行: 每次  $D$  产生变更集  $\Delta$  后, 凡条件  $\kappa_\gamma(\Delta, D)$  成立的  $\gamma$  均应用  $\epsilon_\gamma$ , 无需模型调用。

建模。语义图建模 (SGM) 是算子  $\mathcal{M}_L : (R, P) \mapsto (\Sigma, \mu)$ , 由智能体自身执行, 所用模型  $L$  与随后使用图的模型相同。智能体在模式的分支上编写定义, 验证它们 (引用可解析、键唯一、映射可编译、动作类型正确), 然后发布, 从而重新同步  $D$ 。建模并非独立的离线阶段: 每当请求需要  $\Sigma$  中缺少的概念, 通常是针对新持续指令的自动化规则, 智能体就通过相同的分支-验证-发布周期扩展模式。

与已有对象的关系。引入图后, 记录变为  $d = (A, C \parallel s, G)$ , 其中  $G$  位于外部且共享。Guo (2026c) 的制度关卡, 即模型对每个请求作出的决策, 变为由引擎检查的条件  $c_a$ ; 共享任务空间中的一个区域变为对象集; Guo (2026b) 的闭包集, 即放入上下文的领域知识, 变为按需获取实例的模式。

### 3.3 智能体发生了什么变化

证明见附录 A。第一个命题定位智能体输入的来源, 预测图何时能够节省 token, 何时不能。

命题 1 (输入的构成)。对于保留对话的智能体, 请求  $k$  期间一次模型调用的输入为  $|S| + |p| + \sum_{i < k} |h_i| + |o_k|$ : 系统文本  $S$ 、准备阶段的对话记录  $p$  (研究数据源或图, 或构建图)、先前请求的历史  $h_i$ , 以及当前请求至此产生的工具输出  $o_k$ 。图改变  $o_k$  (以视图替代原始行), 并由此改变后续历史, 也改变  $p$ 。因此, 图只能通过缩小工具输出或缩短准备阶段来减少每次调用的输入 token; 若智能体已能自行计算视图, 例如使用聚合 SQL, 则图无法节省工具输出。依据上下文中数据作答的策略, 对  $D$  中  $n$  行参与的聚合需要  $\Omega(n)$  个输入 token; 若  $m > r$  个条目分别存于独立文本块中, 返回  $r$  个文本块的检索器无法精确回答这些条目的聚合问题。

命题 2 (编译后的合规性)。若  $c_a$  接受时必然满足制度  $\pi(\theta, D)$ , 则  $a$  的每次已提交应用均满足  $\pi$ 。若智能体在上下文中检查  $\pi$ , 每次写入的错误率为  $\epsilon$  且相互独立, 则在  $N$  次写入中至少提交一次违反制度的操作的概率为  $1 - (1 - \epsilon)^N$ 。若  $c_a$  在输入集  $W$  上有误, 则  $W$  中的每次写入都会出错: 编译后的错误是系统性的, 任何共享辅助代码的错误也是如此。

命题 3 (事件处理)。若持续指令由条件和效果均正确的自动化规则实现, 则每个触发事件都能得到处理, 无需模型调用, 因此摘要报告事件的请求不会比未报告事件的请求消耗更多模型调用。自行处理事件的智能体, 对每份包含触发事件的摘要至少付出一次模型调用, 并以其逐事件错误率漏掉各个触发事件。

命题 4 (回本)。设有图和无图时的准备成本分别为  $M_G$  和  $M_0$ , 每请求成本分别为  $c_G$  和  $c_0$ 。若  $c_0 > c_G$  且  $M_G > M_0$ , 累计成本在  $n^* = (M_G - M_0) / (c_0 - c_G)$  个请求后相交; 若  $c_0 \leq c_G < \infty$  且  $M_G > M_0$ , 则永不相交。

命题 1 和 4 (一条核算恒等式) 表明, 图的 token 收益是关于上下文构成的经验问题; 命题 2 和 3 则预测, 通过正确动作写入不会违反制度, 事件也不会增加模型工作。第 6 节测量各项。

## 4 系统

图 1 展示三个部分: 在共享运行时池上运行智能体的运行框架、嵌入工业本体内核的语义图主机, 以及记录系统。

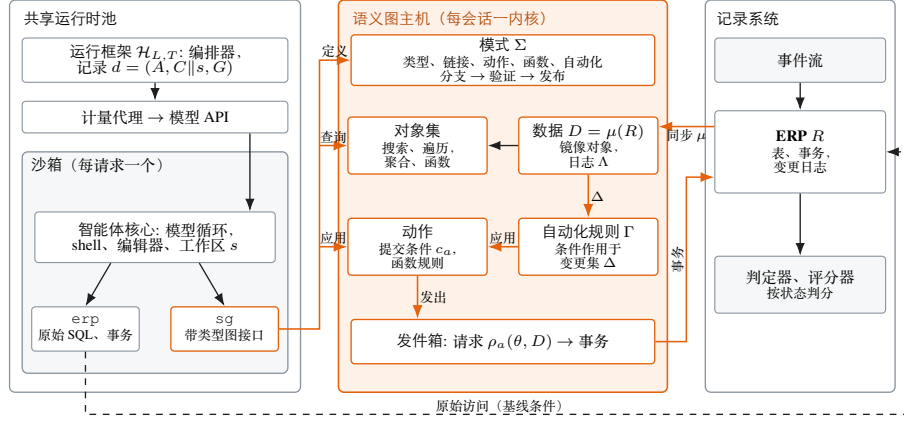


图 1: 架构。智能体核心在每个请求的新沙箱中运行，仅通过带类型的 sg 接口访问图（基线条件使用原始 erp 接口）。主机为每个会话保留一个内核：对象集回答查询，动作检查条件并向发件箱发送请求，发件箱将其作为 ERP 事务执行，产生的变更集  $\Delta$  经  $\mu$  同步回图，并由自动化规则求值。橙色：为语义图建模引入的组件。

**语义图主机。** 主机嵌入生产数据平台的本体内核（版本 0.4.1），实现运营本体服务：模式分支与提案，支持搜索、遍历和聚合的对象集，具有带类型参数、条件和函数规则的动作，隔离函数，以及实时和定时自动化规则。主机增加了三项功能（附录 C）。同步：对象类型由 ERP 表或只读查询支撑；每批事件和每次写入后，主机将 ERP 变更日志应用于镜像只读对象，并让自动化规则对变更集求值。事务发件箱：动作不编辑镜像对象，而是生成 `ErpRequest` 对象，主机按顺序将其作为 ERP 事务执行并同步回来，确保 ERP 始终是唯一的记录系统。能力模式：支持消融阶梯。每个会话都有自己的内存内核。

**带类型接口。** 智能体通过命令行客户端 sg 使用图，主机负责渲染输出并限制其大小：读取命令描述类型，在引擎侧搜索、遍历、计数和聚合，并显示编辑历史；写入命令验证或应用动作；建模命令在分支上定义、映射、验证和发布（附录 B）。基线智能体则使用 erp：对实时 ERP 执行只读 SQL 及其事务。

**共享运行时池上的运行框架。** 智能体在 Guo (2026a) 的运行框架上运行：每个请求是具备 shell 和编辑器的通用编码智能体的一次无状态执行 (Zechner, 2025)，执行前恢复其记录（会话日志和工作区），执行后提交记录；计量代理记录每次模型调用并强制执行支出上限。全部实验均在服务约 290 个生产智能体的运行时池上进行，而非使用专用硬件；每次执行在隔离且受资源限制的沙箱中运行，只能看到语言运行时、两个客户端及自身工作区（附录 C）。

## 5 OpsWEEK：连续企业运营基准

现有企业工作智能体基准要么在每个任务后重置世界 (Styles et al., 2024; Yao et al., 2025)，要么仅以最终结果评估长任务 (Xu et al., 2025a)。两者都未测试 SGM 所针对的场景：在同一个不断演化的记录系统上，依据书面制度处理多个请求，并执行适用于未来事件的持续指令。OpsWEEK 生成这样的会话。<sup>2</sup>所有元素均为合成，且由随机种子确定；公司名称为 Demo Company。

**记录系统。** 一个基于 SQLite 的传统 ERP，包含十三张表（客户、物料、供应商、仓库、库存、销售订单及明细、发运、发票、采购订单及明细、价格、通知），另有一张导入垃圾数据表。名称遵循 ERP 惯例 (`SO_HDR`, `STAT`, `INV_BAL`, `QTY_RSV`)，值以编码表示。写入通过十三种事务进行（创建销售订单、变更其状态或明细、分配订单、创建采购订单、变更价格、设置信用状态、通知团队、...）。这些事务与真实 ERP 一样，强制保证完整性，但不强制执行业务制度。

<sup>2</sup>OpsWEEK 与评估 IT 运维的 OpsBench (Heyno, 2026) 和 Cloud-OpsBench (Wang et al., 2026c) 无关，也与 OpsWorld 协议 (ColabHive Research, 2026) 无关。

文档。数据字典以简略的 ERP 风格解释约 70% 的列。六份制度文档（信用、折扣与审批、优先级与分配、补货、订单变更、通知）用业务语言规定规则；某些编码仅在其中解释（战略客户指位于两个指定地区的金牌客户）。规则相互作用：信用冻结优先于审批；审批阈值适用于折扣后的订单金额；可用信用额度需扣除未结清余额及所有未结订单金额。

事件与会话。一个会话模拟一个工作周：先有一个准备请求，随后五天，每天八个请求。每天开始前及每两个请求后，一批事件改变 ERP：EDI 订单到达、已放行订单发运并开票、采购订单收货、发票付款、供应商成本与客户信用发生变化。生成器保证每天都有触发信用和审批规则的订单，以及库存低于再订货点的物料。智能体在下一个请求的开头看到每批事件的一行摘要（数量与标识符）。

请求。九类请求包含二十二种模板（附录表 6）：查找、多跳问题、数百至数千行上的聚合、条件分析、受制度约束的写入（录入带折扣的电话订单；修改、取消、审批和分配订单）、批量更新、关于智能体自身先前工作的问题、报告，以及四条持续指令：库存低于再订货点时自动补货；每笔新 EDI 订单到达时，按信用和审批规则检查；战略客户的优先级为 1 的订单一旦处于未结状态即分配；第四天将审批阈值从 50,000 降至 40,000。参数沿参考轨迹采样，确保每个请求都有实际意义，每次回复均以 ANSWER: <json> 结束。

判分。判分按状态进行：参考判定器在智能体自身的请求前状态上计算预期答案和预期变更，因此早期错误不会在后续每个答案中重复计错。写入依据请求时间窗口内的 ERP 变更日志，按每行的净变化判分：每项预期变更都必须存在且字段正确，任何变更都不得涉及请求范围外的行；自由文本字段、订单明细的顺序，以及智能体在请求内纠正的中间状态不影响判分。数值容差为 0.5%（计数需精确相等），集合必须完全一致，编码及其文档说明的含义均可接受（附录 G）。请求成功率按 36 个非持续指令请求计算；指令本身只需确认收到，并在其后的每个检查点（每批事件后和每个请求后）判分：每个触发事件的效果必须在下一个请求结束前出现，重复效果或无触发事件的效果均计为错误。若请求的前提此前已被智能体自身破坏（它取消了当前要求审批的订单），则标记为前提被破坏并计为失败；我们将其单独报告为错误传播。应用参考判定器操作的脚本化智能体在每个生成会话上均得分 100%；另一个仅通过专家图工作（附录 E）的脚本化智能体能解决参考会话。这验证了生成器、参考判定器和评分器一致，并不意味着每份制度文本对读者都无歧义。

规模。三种规模改变数据量而保持会话不变：S（24 个客户、60 种物料、约 300 笔历史订单；序列化数据为 28k 个 token）、M（120 个客户、400 种物料、约 2,000 笔订单；204k 个 token）及 L（600 个客户、2,000 种物料、约 15,000 笔订单；1454k 个 token）。生成器、ERP 服务器、参考判定器和评分器可向作者获取。

## 6 实验

### 6.1 设置

条件。七个条件使用相同模型、请求文本、事件摘要、答案格式和 ERP 事务；区别在于智能体如何读取和写入（表 1）。LLM 在每个请求中收到 CSV 格式的完整当前 ERP；RAG 收到混合检索器返回的前 30 个块，检索器将 BM25 与稠密编码器（Robertson & Zaragoza, 2009; Xiao et al., 2024）通过倒数排名融合，检索行、表和文档块，每次变更后重建索引。两者每请求仅调用一次模型，是下限基线，并非使用工具的竞争方案。Agent 是具备 shell、工作周内持久工作区和原始 erp 接口（只读 SQL 和事务）的编码智能体。KG、+Actions 和 SGM-expert 为同一智能体提供能力逐级增加的专家图，不提供 SQL（RQ3）；SGM-self 提供只读 SQL 和建模命令，不提供原始事务，因此每次写入都通过其自行定义的动作执行。每个智能体条件均以相同预算（120 次模型调用、40 分钟）的准备请求开始：原始智能体准备笔记和辅助脚本，图条件研究图，SGM-self 按六步步骤说明构建图（附录 D）。专家图（附录 E）由作者依据制度并在了解请求模板的情况下编写，是乐观参考而非上界；SGM-self 从未看到它。

设计。主研究使用 M 规模，每会话四十个请求；每个种子对应一家不同公司，同一种子上的所有条件看到相同请求和事件，因此请求按文本配对（并非按状态，因为判分依赖状态）。Agent、SGM-expert 和 SGM-self 在 6 个种子上运行，RAG、KG 和 +Actions 在三个种子上运行，成本最高的 LLM 在一个种子上运行；对比使用两个条件共有的种子。每请求最多 30 次模型调用和 10 分钟。超限请求计为失败并保留成本，会话继续（共 3 次执行：KG 的一次准

表 1: 实验条件。全部使用 DeepSeek V4.1 Flash、低推理强度及提供商默认采样设置，并使用相同请求、摘要和事务。

| 条件                 | 读取                  | 写入    | 持续规则      | 跨请求状态    |
|--------------------|---------------------|-------|-----------|----------|
| LLM                | 提示中的完整 ERP 快照       | 事务列表  | 由模型处理     | 对话       |
| RAG                | 检索所得前 30 个块         | 事务列表  | 由模型处理     | 对话       |
| Agent (L0, 原始 ERP) | SQL                 | 原始事务  | 由智能体处理    | 对话 + 工作区 |
| KG (L1)            | 图 (无 SQL): 搜索、遍历、聚合 | 原始事务  | 由智能体处理    | 对话 + 工作区 |
| +Actions (L2)      | 图 (无 SQL)           | 动作    | 由智能体处理    | 对话 + 工作区 |
| SGM-expert (L3)    | 图 (无 SQL) + 函数      | 动作    | 自动化规则或智能体 | 对话 + 工作区 |
| SGM-self           | 自建图 + SQL           | 自定义动作 | 自动化规则或智能体 | 对话 + 工作区 |

表 2: M 规模的主要结果。成功率基于已判分的非指令请求，附 95% Wilson 区间；每请求 token、成本和时间基于全部请求；SR 召回率：及时满足的持续规则义务比例（重新评分，附录 G）；准备：智能体条件的首个请求。

| 条件           | 种子数 / 判分请求 | 成功率 [95% CI] (%)    | 令牌/请求 (k) | 元/请求  | 耗时中位数 (s) | 模型时间均值 (s) | 调用/请求 | 常驻规则召回率 (%) | 准备成本 (元) |
|--------------|------------|---------------------|-----------|-------|-----------|------------|-------|-------------|----------|
| 长上下文 LLM     | 1 / 36     | 77.8 [61.9, 88.3]   | 446.4     | 1.003 | 20.4      | 41.3       | 1.00  | 63.3        | 0.00     |
| RAG          | 3 / 108    | 24.1 [17.0, 32.9]   | 10.2      | 0.016 | 4.8       | 5.1        | 1.00  | 4.6         | 0.00     |
| 智能体 (原始 ERP) | 6 / 216    | 97.7 [94.7, 99.0]   | 311.2     | 0.027 | 11.2      | 8.7        | 3.62  | 96.4        | 0.23     |
| 知识图谱 (L1)    | 3 / 108    | 100.0 [96.6, 100.0] | 348.2     | 0.034 | 11.2      | 11.4       | 3.97  | 99.7        | 0.12     |
| + 动作 (L2)    | 3 / 108    | 98.1 [93.5, 99.5]   | 269.8     | 0.024 | 9.6       | 8.4        | 3.57  | 90.5        | 0.13     |
| SGM-专家图 (L3) | 6 / 215    | 99.1 [96.7, 99.7]   | 344.9     | 0.031 | 9.0       | 11.0       | 3.69  | 92.6        | 0.15     |
| SGM-自建       | 6 / 215    | 99.5 [97.4, 99.9]   | 514.4     | 0.037 | 9.9       | 10.7       | 3.21  | 95.7        | 0.82     |

备，以及 SGM-expert 和 SGM-self 各一个请求)；遭遇有记录的基础设施故障的请求被排除 (附录 G)。提示与接口帮助在一个 S 规模试运行会话上调试，该会话不纳入结果。

**判分与统计。** 主要结果指标是请求成功率。三个预先设定的对比为 SGM-self 对 Agent、SGM-expert 对 Agent，以及 SGM-self 对 SGM-expert。我们对配对请求采用 McNemar 检验及 Holm 校正，并对每种子的差异采用精确符号翻转检验，因为同一会话内的请求相互依赖。运行结束后，我们对所有条件同等修正判分规则：接受数字格式差异和编码的文档含义，按最终状态比较写入，从触发时刻开始计算持续规则义务 (附录 G)；因此，对比预先设定，但事后判分。成功率区间采用 Wilson 区间；差异和比值也在种子层面报告。成本分别按标价（每百万未缓存输入 token 0.30 USD、缓存输入 0.006 USD、输出 1.20 USD；每 USD 为 7.2 CNY）及将全部输入 token 按未缓存价格计费计算。速度采用每请求挂钟时间中位数和模型时间均值。规模研究在 S 规模（五个条件）和 L 规模（原始智能体和 SGM-self）重复一个会话的前二十个请求，迁移研究则在 60 个 WorkBench 任务上运行原始智能体和 SGM-self (Styles et al., 2024)。

## 6.2 RQ1: 准确率、TOKEN 和时间

**准确率。** 所有智能体条件均接近上限 (表 2): SGM-self 的请求成功率为 99.5%，SGM-expert 为 99.1%，原始智能体为 97.7%，只读图和 +Actions 分别为 100.0 和 98.1%。SGM-self 比原始智能体高 1.9 个百分点（不一致请求为 5 对 1；McNemar  $p = 0.219$ ，Holm  $p = 0.656$ ）；种子层面的差异为 1.8 个百分点 (95% CI [-2.2, 5.8]，6 个种子的符号翻转检验  $p = 0.500$ )。另外两个对比也不显著 (Holm  $p = 0.906$  和 1.000)。结果无定论，不能视为没有差异：在接近上限的水平上，本研究只能检测较大差异。原始智能体的失败全部出现在聚合、条件查询和报告中 (附录表 7)。下限基线远远落后：长上下文提示成功率为 77.8%，每请求成本 1.003 CNY；检索成功率为 24.1%。

**Token、成本和时间。** 图末节省 token。SGM-self 每请求 token 数为原始智能体的  $1.66\times$  (种子层面  $1.58\times$ ，95% CI [0.99, 2.52]，各种子范围 0.93–3.02)，SGM-expert 为  $1.11\times$  (种子层面  $1.10\times$  [0.89, 1.38])。由于智能体输入 token 的 99.7–99.8% 命中缓存，SGM-self 的标价成本仅为  $1.38\times$  (种子层面  $1.32\times$  [0.88, 1.98])，每请求多 0.010 CNY；按未缓存价格，每请求成本为 0.681 (Agent) 和 1.121 CNY (SGM-self)。输入构成与命题 1 一致 (附录表 4)：所有智能体条件的工具输出都较小 (原始智能体每请求 1203 个字符，其聚合 SQL 已经返回视图)，而每次调用都重新发送准备阶段的对话记录，SGM-self 为 454k 字符，原始智能体为 149k，因此每次调用的输入分别为 159.6k 和 88.1k token。时间差异没有一致方向：SGM-self 调用更少 (3.21 对 3.62)，挂钟时间中位数更短 (9.9 对 11.2 s)，但模型时间为  $1.22\times$ 。

持续规则与写入。图条件中的自动化规则接管了持续规则工作：SGM-self 的自动化规则无需模型调用便满足了 78.6% 的义务。召回率未提高：及时满足的义务比例，原始智能体为 96.4%，SGM-expert 为 92.6%，SGM-self 为 95.7%（附录表 5）。大多数遗漏涉及同一规则（战略客户优先级为 1 的订单一旦处于未结状态就分配），以及下达规则时已处于未结状态的订单；智能体对这些订单是否属于“从现在起”的范围有不同解读。摘要将事件推送给智能体，因此所有条件均无需轮询，也未奖励即时响应。原始智能体在摘要报告事件的请求中，比无事件请求多用 1.02 次模型调用；SGM-expert 的差异为 -0.33，SGM-self 为 0.24（未按模板调整），方向与命题 3 一致。所有智能体条件几乎在每个写入请求中都留下正确的最终状态（附录图 2）；同样用原始事务写入的原始智能体和只读图，每会话分别有 0.33 次和 0.67 次临时违反制度（创建订单时使用制度禁止的状态，并在同一请求内修正），动作条件没有。没有请求要求执行制度禁止的写入，因此命题 2 的条件从未需要拒绝此类写入。

### 6.3 RQ2: 自建图

智能体构建了什么。在准备请求中（7.3 分钟，范围 4.7–11.3；0.82 CNY，比原始智能体的准备多 0.59），SGM-self 平均构建了 14.7 个对象类型、13.8 个链接类型、15.3 个动作、18.2 个函数及 4.2 个自动化规则。它映射了专家图映射的每张源表，但采用了不同设计：专家图中 48.3% 的映射属性和 60.6% 的链接在自建图中有对应项（附录表 8）。它在动作的函数规则中编码信用、折扣、审批和分配制度，仅依据制度文本就为新 EDI 订单和补货建立自动化规则，并在工作周内继续建模（每会话发布 4.7 个新模型版本）。

如何使用图，以及原始智能体的做法。SGM-self 仅通过自己的动作写入，并让自动化规则处理事件，但它通过 SQL 读取：75.9% 的读取命令是原始 SQL。原始智能体也按准备指令的建议，在没有图的情况下编译制度：在 6 个会话中的 5 个会话里，它在准备阶段编写了辅助脚本库（函数用于依据信用与审批规则评估订单、处理新 EDI 订单或选择补货候选项），每会话运行 52 次。两个智能体也都依据制度文本，在收到要求前就按规则行动（原始智能体产生 28 个效果，SGM-self 产生 16 个）。因此，两者的区别是领域如何被编译：编译为引擎强制执行的带类型动作及由引擎触发的自动化规则，或编译为智能体记得运行的脚本。

成本与回本。保留对话时，SGM-self 的每请求成本高于原始智能体，因此额外准备成本从未回收（命题 4）。由于额外输入来自重复发送的准备阶段对话记录，我们在看到这些结果后决定运行一个探索性变体：准备后清空对话，工作周开始时仅保留工作区，图条件还保留图（种子 1–3）。清空对话后，SGM-self 每请求 token 数降为原来的  $0.53 \times$ （95% CI [0.46, 0.62]），原始智能体降为  $0.66 \times$ ，成功率不变（分别为 100.0% 和 97.2%；附录表 9）。两者的差距缩小但没有消失：清空对话后，SGM-self 的 token 数仍为原始智能体的  $1.27 \times$  [1.11, 1.46]，调用次数更多（4.02 对 3.80），每次调用的上下文也更大；其累计成本仍未低于原始智能体。

### 6.4 RQ3: 图的哪部分起作用

阶梯基于专家图，以图读取替代 SQL；最后一级同时增加函数和自动化规则，无法分离两者的效果。原始智能体、只读图、+Actions 和 SGM-expert 的查询成功率分别为 96.7、100.0、98.7 和 99.3%，写入成功率分别为 100.0、100.0、97.0 和 98.5%（附录图 2）；任何一级与原始智能体的差异都不超过几个请求。成本同样不确定：在种子层面，只读图的每请求成本为原始智能体的  $1.28 \times$ （95% CI [0.99, 1.67]），+Actions 为  $0.92 \times$  [0.64, 1.32]，每请求调用数与原始智能体相当（3.57 对 3.62）。各级改变的仍是工作方式：从 +Actions 起，写入通过经过验证的动作执行；从 SGM-expert 起，自动化规则满足了 74.1% 的持续规则义务。

### 6.5 规模与迁移

原始智能体的每请求成本没有随数据增长：在一个会话的前二十个请求中，它在 S 规模（数据 28k 个 token）每请求使用 261.3k token，在 L 规模（1454k）使用 241.5k，成功率分别为 100.0% 和 100.0%。SGM-self 的成功率为 100.0% 和 100.0%，每请求 token 数为 406.0k 和 884.1k；它的 token 数跟随重复发送的准备阶段对话记录变化，而不是跟随数据：它在 S 规模的准备超过时间限制，而在 L 规模其准备记录达到 481k 字符（附录图 4）。长上下文提示的成功率从 S 规模的 100.0%（每请求 0.158 CNY）降到 M 规模的 77.8%，L 规模的数据则放不进它的上下文。WorkBench 没有制度，每任务前重置数据；在 60 个任务上，原始智能体成功率为 96.7%，SGM-self 为 88.3%，每任务 token 数分别为 408.4k 和 322.1k，时间分别为 28.0 和 11.4s；SGM-self 还可以通过 SQL 读取表，而原始智能体的 WorkBench 工具不提

供这一能力。SGM-self 的多数失败来自跨两个领域的任务：它将“next Friday”解读为即将到来的周五，并在任务间沿用这一解读。

## 7 讨论与结论

图改变了什么。在我们的设置中，图未使智能体的准确率、成本或速度得到可检测的改善。它改变了工作方式：写入通过由引擎检查的动作执行，持续指令作为自动化规则运行，无需模型调用。原始智能体将相同制度编译为自行运行的辅助脚本库，因此图特有的贡献是强制执行与触发器，而非编译后的知识。命题 1 指出图何时能够节省 token：工具输出占据输入的主要部分时（例如智能体没有查询语言），以及准备阶段的对话记录移出上下文时（第 6.3 节）。

自主建模与治理。两类智能体都依据制度文本执行了尚未有人要求的规则；组织必须决定是否允许智能体将制度文本转为执行规则，而图至少使这些规则可见且有版本记录。

表示还是能力？我们的阶梯在同一个领域模型上增加能力，并未将模型与能力分离。原始智能体若配备带制度检查的事务包装器 (Zwerdling et al., 2025; Wang et al., 2026a)，或数据库上的触发器运行时，或许无需带类型的图就能获得相同的强制执行和事件处理能力；我们的原始智能体已经编写了此类包装器会运行的制度代码。带类型的模型是否正是智能体能够编写正确守卫和触发器的原因，仍是开放问题。

局限。OpsWEEK 是合成基准：只有一种公司类型，制度文本清晰，参考判定器在可能存在多种解读时固定采用一种。会话含四十个请求，事件通过摘要推送，没有请求必须拒绝，第四天的阈值变更只影响少量订单，0.5% 的容差可能接受遗漏小额订单的聚合结果；基准未充分检验编译后的条件，也不奖励即时响应。请求按文本而非状态配对。专家图是在了解请求模板的情况下编写的，因此 SGM-expert 是乐观参考。我们仅测试了一个采用低推理强度的低成本模型；LLM 和 RAG 是单次调用下限基线。推断基于三个主要条件各六个会话，KG、+Actions 和 RAG 各三个，以及 LLM 的一个会话，无法检测几个百分点的差异；判分在运行结束后修正（附录 G），清空对话变体属于探索性分析。本体内核属于商业平台，未发布。这些局限也决定了下一步工作：在 OpsWEEK 中加入必须拒绝的请求、更长的工作跨度和噪声更多的数据，并把图与不使用图的守卫和触发器运行时进行比较 (Guo, 2026d)。

结论。低成本模型在几分钟内构建的可执行知识图谱，未使准确率产生可检测的变化，并消耗更多 token；它改变了一周工作的完成方式：制度检查移入由引擎检查的动作，持续指令移入自动化规则。

## 参考文献

- Dean Allemang and Juan Sequeda. Increasing the Accuracy of LLM Question-Answering Systems with Ontologies. In *The Semantic Web – ISWC 2024*, pp. 324–339, 2024. doi: 10.1007/978-3-031-77847-6\_18. URL [https://link.springer.com/10.1007/978-3-031-77847-6\\_18](https://link.springer.com/10.1007/978-3-031-77847-6_18).
- Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 12–20, 2025. doi: 10.24963/ijcai.2025/2. URL <https://www.ijcai.org/proceedings/2025/2>.
- Anthropic. Model Context Protocol Specification. <https://modelcontextprotocol.io>, 2024. Open protocol specification, accessed 2026-09-23.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://arxiv.org/abs/2310.11511>. arXiv:2310.11511.
- Hamed Babaei Giglou, Jennifer D’Souza, and Sören Auer. LLMs4OL: Large Language Models for Ontology Learning. In *The Semantic Web – ISWC 2023*, pp. 408–427, 2023.

- doi: 10.1007/978-3-031-47240-4\_22. URL [https://link.springer.com/10.1007/978-3-031-47240-4\\_22](https://link.springer.com/10.1007/978-3-031-47240-4_22).
- Axel Backlund and Lukas Petersson. Vending-Bench: A Benchmark for Long-Term Coherence of Autonomous Agents. arXiv preprint arXiv:2502.15840, 2025. URL <https://arxiv.org/abs/2502.15840>.
- Jinheon Baek, Alham Aji, and Amir Saffari. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pp. 70–98, 2023. doi: 10.18653/v1/2023.matching-1.7. URL <https://aclanthology.org/2023.matching-1.7>.
- Jiaxin Bai, Wei Fan, Qi Hu, Qing Zong, Chunyang Li, Hong Ting Tsang, Hongyu Luo, Yauwai Yim, Haoyu Huang, Xiao Zhou, Feng Qin, Tianshi Zheng, Xi Peng, Xin Yao, Huiwen Yang, Leijie Wu, Ji Yi, Gong Zhang, Renhai Chen, and Yangqiu Song. AutoSchemaKG: Autonomous Knowledge Graph Construction through Dynamic Schema Induction from Web-Scale Corpora. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20557–20584, 2026. doi: 10.18653/v1/2026.acl-long.942. URL <https://aclanthology.org/2026.acl-long.942>.
- François Bancilhon and Nicolas Spyratos. Update Semantics of Relational Views. *ACM Transactions on Database Systems*, 6(4):557–575, 1981. doi: 10.1145/319628.319634. URL <https://doi.org/10.1145/319628.319634>.
- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan.  $\tau^2$ -Bench: Evaluating Conversational Agents in a Dual-Control Environment. arXiv preprint arXiv:2506.07982, 2025. URL <https://arxiv.org/abs/2506.07982>.
- Kamal Bhattacharya, Nathan S. Caswell, Santhosh Kumaran, Anil Nigam, and Frederick Y. Wu. Artifact-centered operational modeling: Lessons from customer engagements. *IBM Systems Journal*, 46(4):703–721, 2007. doi: 10.1147/SJ.464.0703. URL <https://doi.org/10.1147/SJ.464.0703>.
- Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. WorkArena++: Towards Compositional Planning and Reasoning-based Common Knowledge Work Tasks. In *Advances in Neural Information Processing Systems 37*, pp. 5996–6051, 2024. doi: 10.52202/079017-0195. URL <http://www.proceedings.com/079017-0195.html>.
- Pranav Bykampadi, Neel Mokaria, Vishesh Narayan, Faizan Wajid, and Ashok Agrawala. From Extraction to Governed Memory: Multi-Agent Knowledge Graph Construction with Domain-Expert Review. arXiv preprint arXiv:2608.28642, 2026. URL <https://arxiv.org/abs/2608.28642>.
- Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large Language Models as Tool Makers. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.17126.
- Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487, 2016. doi: 10.3233/sw-160217. URL <https://journals.sagepub.com/doi/full/10.3233/SW-160217>.
- Peter Baile Chen, Devin Yang, Weiyue Li, Fabian Wenz, Yi Zhang, Nesime Tatbul, Michael Cafarella, Çağatay Demiralp, and Michael Stonebraker. BEAVER: An Enterprise Benchmark for Text-to-SQL. arXiv preprint arXiv:2409.02038, 2024. URL <https://arxiv.org/abs/2409.02038>.
- Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding Agents via Verifiable Safety Policy Reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8313–8344. PMLR, 2025. URL <https://proceedings.mlr.press/v267/chen25ae.html>.

- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv preprint arXiv:2504.19413*, 2025. URL <https://arxiv.org/abs/2504.19413>.
- Meiduo Chong, Shaolei Zhang, Ju Fan, and Xiaoyong Du. EvoOntology: A Self-Evolving Ontology Layer for Data Agents. *arXiv preprint arXiv:2609.15779*, 2026. URL <https://arxiv.org/abs/2609.15779>.
- ColabHive Research. Monolith, Hive, and Society: R6 Protocol v3. ColabHive Research protocol, September 2026, 2026. URL <https://colabhive.com/research/monolith-hive-society.html>. Design protocol with zero live R6 measurements; verified from indexed publisher page; accessed 2026-10-08.
- Cube. Cubes. Cube Documentation, 2026. URL <https://docs.cube.dev/reference/data-modeling/cube>. Undated living documentation; year denotes consulted version; accessed 2026-10-08.
- Das. R2RML: RDB to RDF Mapping Language. W3C Recommendation, 2012. URL <https://www.w3.org/TR/r2rml/>. Accessed 2026-10-08.
- Umeshwar Dayal and Philip A. Bernstein. On the Correct Translation of Update Operations on Relational Views. *ACM Transactions on Database Systems*, 7(3):381–416, 1982. doi: 10.1145/319732.319740. URL <https://doi.org/10.1145/319732.319740>.
- dbt Labs. dbt Semantic Layer. dbt Developer Hub documentation, 2026. URL <https://docs.getdbt.com/docs/use-dbt-semantic-layer/dbt-sl>. Last updated September 29, 2026; accessed 2026-10-08.
- Giuseppe De Giacomo, Domenico Lembo, Xavier Oriol, Domenico Fabio Savo, and Ernest Teniente. Practical Update Management in Ontology-Based Data Access. In *The Semantic Web – ISWC 2017*, volume 10587 of *Lecture Notes in Computer Science*, pp. 225–242. Springer, 2017. doi: 10.1007/978-3-319-68288-4\_14. URL [https://link.springer.com/chapter/10.1007/978-3-319-68288-4\\_14](https://link.springer.com/chapter/10.1007/978-3-319-68288-4_14).
- Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How Capable Are Web Agents at Solving Common Knowledge Work Tasks? In *International Conference on Machine Learning (ICML)*, 2024. URL <https://arxiv.org/abs/2403.07718>. arXiv:2403.07718.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL <https://arxiv.org/abs/2404.16130>.
- Shengda Fan, Xin Cong, Yuepeng Fu, Zhong Zhang, Shuyan Zhang, Yuanwei Liu, Yesai Wu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. WorkflowLLM: Enhancing Workflow Orchestration Capability of Large Language Models. In *International Conference on Learning Representations*, pp. 24498–24525, 2025. URL [https://proceedings.iclr.cc/paper\\_files/paper/2025/hash/3d7259031023c5aa463187c4a31c95c8-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2025/hash/3d7259031023c5aa463187c4a31c95c8-Abstract-Conference.html).
- Martin Fowler. Event Sourcing. MartinFowler.com, 12 December 2005, 2005. URL <https://martinfowler.com/eaDev/EventSourcing.html>. Accessed 2026-10-08.
- Martin Fowler. CQRS. MartinFowler.com, 14 July 2011, 2011. URL <https://martinfowler.com/bliki/CQRS.html>. Accessed 2026-10-08.
- Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *Proceedings of the VLDB Endowment*, 17(5):1132–1145, 2024. doi: 10.14778/3641204.3641221. URL <https://dl.acm.org/doi/10.14778/3641204.3641221>.

- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 10764–10799, 2023a. URL <https://proceedings.mlr.press/v202/gao23f.html>.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997*, 2023b.
- Google. Introduction to LookML. Google Cloud Looker documentation, 2026. URL <https://docs.cloud.google.com/looker/docs/what-is-lookml>. Last updated October 6, 2026; accessed 2026-10-08.
- Benjamin Gruenbaum, Doron Porat, Assaf Natanzon, Roy Zavida, Chen Dinachi, and Or Itzhary. The Era by Eon Benchmark: A Generated Enterprise Estate with Exact Ground Truth for Benchmarking LLM Agents. *arXiv preprint arXiv:2609.09853*, 2026. URL <https://arxiv.org/abs/2609.09853>.
- Michael Grüninger and Mark S. Fox. Methodology for the Design and Evaluation of Ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95*, Montreal, Canada, 1995. URL <https://www.eil.utoronto.ca/wp-content/uploads/enterprise-modelling/papers/gruninger-ijcai95.pdf>.
- Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging Pre-trained Large Language Models to Construct and Utilize World Models for Model-based Task Planning. In *Advances in Neural Information Processing Systems 36*, pp. 79081–79094, 2023. doi: 10.52202/075280-3459. URL <http://www.proceedings.com/075280-3459.html>.
- Yanming Guo. AIDH: An Agent-Independent Distributed Harness for Persistent Agents. Manuscript, 2026a.
- Yanming Guo. Closure instead of training: Agentic domain modelling for life cycle assessment. Manuscript in preparation, The University of Sydney, 2026b.
- Yanming Guo. Neural orchestration: Spatio-temporal communication for large-scale multi-agent systems. Working paper, The University of Sydney, 2026c.
- Yanming Guo. Recursive Self-Improvement in Self-Organizing Multi-Agent Systems. Working paper, The University of Sydney, 2026d.
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and Fast Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, 2025. doi: 10.18653/v1/2025.findings-emnlp.568. URL <https://aclanthology.org/2025.findings-emnlp.568>.
- Bernal Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *Advances in Neural Information Processing Systems 37*, pp. 59532–59569, 2024. doi: 10.52202/079017-1902. URL <http://www.proceedings.com/079017-1902.html>.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In *International Conference on Machine Learning*, 2025. URL <https://arxiv.org/abs/2502.14802>.
- Kate Gwimm and Carson Eisenach. Beyond the Harness: End-to-End Optimization of Context Artifacts for Enterprise Text-to-SQL. *arXiv preprint arXiv:2608.22830*, 2026. URL <https://arxiv.org/abs/2608.22830>.
- Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann Lecun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In *Advances in Neural Information Processing Systems 37*, pp. 132876–132907, 2024. doi: 10.52202/079017-4224. URL <http://www.proceedings.com/079017-4224.html>.

- Heyno. Introducing OpsBench. Heyno research announcement, 30 June 2026, 2026. URL <https://www.heyno.net/opsbench>. Official announcement; internal results and planned external availability; accessed 2026-10-08.
- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D’amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *ACM Computing Surveys*, 54(4):1–37, 2021. doi: 10.1145/3447772. URL <https://dl.acm.org/doi/10.1145/3447772>.
- Kelly Hong, Anton Troynikov, and Jeff Huber. Context Rot: How Increasing Input Tokens Impacts LLM Performance. Technical report, Chroma, 2025. <https://research.trychroma.com/context-rot>.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Krizan, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What’s the Real Context Size of Your Long-Context Language Models? In *Conference on Language Modeling*, 2024. URL <https://arxiv.org/abs/2404.06654>.
- Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. ChatDB: Augmenting LLMs with Databases as Their Symbolic Memory. arXiv preprint arXiv:2306.03901, 2023. URL <https://arxiv.org/abs/2306.03901>.
- Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems. In *International Conference on Learning Representations (ICLR)*, 2025. URL <https://arxiv.org/abs/2408.08435>. arXiv:2408.08435.
- Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. In *International Conference on Learning Representations*, pp. 156259–156291, 2026. URL [https://proceedings.iclr.cc/paper\\_files/paper/2026/hash/fd1eff9dd295df50a41f2521942fa31d-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2026/hash/fd1eff9dd295df50a41f2521942fa31d-Abstract-Conference.html).
- Kung-Hsiang Huang, Akshara Prabhakar, Sidharth Dhawan, Yixin Mao, Huan Wang, Silvio Savarese, Caiming Xiong, Philippe Laban, and Chien-Sheng Wu. CRMarena: Understanding the Capacity of LLM Agents to Perform Professional CRM Tasks in Realistic Environments. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3830–3850, 2025. doi: 10.18653/v1/2025.naacl-long.194. URL <https://aclanthology.org/2025.naacl-long.194>.
- Kung-Hsiang Huang, Akshara Prabhakar, Onkar Thorat, Divyansh Agarwal, Prafulla Kumar Choubey, Yixin Mao, Silvio Savarese, Caiming Xiong, and Chien-Sheng Wu. CRMarena-Pro: Holistic Assessment of LLM Agents Across Diverse Business Scenarios and Interactions. *Transactions on Machine Learning Research*, 2026. URL <https://openreview.net/pdf/51fa206d5000500ead8a5b13b49b690c40616329.pdf>.
- Richard Hull, Elio Damaggio, Riccardo De Masellis, Fabiana Fournier, Manmohan Gupta, Fenno F. Terry Heath III, Stacy Hobson, Mark H. Linehan, Sridhar Maradugu, Anil Nigam, Piyawadee Noi Sukaviriya, and Roman Vaculín. Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events. In *Proceedings of the 5th ACM International Conference on Distributed Event-Based Systems*, pp. 51–62, 2011. doi: 10.1145/2002259.2002270. URL <https://doi.org/10.1145/2002259.2002270>.
- Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9237–9251, 2023. doi: 10.18653/v1/2023.emnlp-main.574. URL <https://aclanthology.org/2023.emnlp-main.574>.
- Ziyan Jiang, Xueguang Ma, and Wenhui Chen. LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs. arXiv preprint arXiv:2406.15319, 2024. URL <https://arxiv.org/abs/2406.15319>.

- Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI Agents That Matter. *arXiv preprint arXiv:2407.01502*, 2024. URL <https://arxiv.org/abs/2407.01502>.
- Knublauch. Shapes Constraint Language (SHACL). W3C Recommendation, 20 July 2017, 2017. URL <https://www.w3.org/TR/2017/REC-shacl-20170720/>. Accessed 2026-10-08.
- Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. iText2KG: Incremental Knowledge Graphs Construction Using Large Language Models. In *Web Information Systems Engineering – WISE 2024*, pp. 214–229, 2024. doi: 10.1007/978-981-96-0573-6\_16. URL [https://link.springer.com/10.1007/978-981-96-0573-6\\_16](https://link.springer.com/10.1007/978-981-96-0573-6_16).
- Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows. In *International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2411.07763>.
- Maurizio Lenzerini. Ontology-based data management. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 5–6, 2011. doi: 10.1145/2063576.2063582. URL <https://dl.acm.org/doi/10.1145/2063576.2063582>.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL <https://arxiv.org/abs/2005.11401>. arXiv:2005.11401.
- Jinyang Li, Binyuan Hui, Ge Qu, Jiayi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Ma Chenhao, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM Already Serve as A Database Interface? A Big Bench for Large-Scale Database Grounded Text-to-SQLs. In *Advances in Neural Information Processing Systems 36*, pp. 42330–42357, 2023. doi: 10.52202/075280-1835. URL <http://www.proceedings.com/075280-1835.html>.
- Ruizhe Li, Licheng Zhang, Benfeng Xu, Mingxuan Du, Zheren Fu, and Weidong Chen. When Your Agent Opens the Chat App: Agent-Controlled Search over Raw Chat Logs Rivals Structured Memory. *arXiv preprint arXiv:2608.12888*, 2026. URL <https://arxiv.org/abs/2608.12888>.
- Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. StructRAG: Boosting Knowledge Intensive Reasoning of LLMs via Inference-time Hybrid Information Structurization. In *International Conference on Learning Representations*, pp. 36107–36124, 2025. URL [https://proceedings.iclr.cc/paper\\_files/paper/2025/hash/5975754c7650dfec0682e06elfec0522-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2025/hash/5975754c7650dfec0682e06elfec0522-Abstract-Conference.html).
- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 881–893, 2024. doi: 10.18653/v1/2024.emnlp-industry.66. URL <https://aclanthology.org/2024.emnlp-industry.66>.
- Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as Policies: Language Model Programs for Embodied Control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2023. doi: 10.1109/icra48891.2023.10160591. URL <https://ieeexplore.ieee.org/document/10160591/>.
- Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, Wen Zhang, and Huajun Chen. KAG: Boosting

- LLMs in Professional Domains via Knowledge Augmented Generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 334–343, 2025. doi: 10.1145/3701716.3715240. URL <https://dl.acm.org/doi/10.1145/3701716.3715240>.
- Chengxi Liao, Tao Xu, Zulong Chen, Chuanfei Xu, Yiyan Wang, Xinyun Wang, Yanlong Zhang, Xiaojun Chen, Zhibo Yang, and Zeyi Wen. EntSQL: A Benchmark for Grounding Text-to-SQL in Long-Context Enterprise Knowledge. arXiv preprint arXiv:2606.03363, 2026. URL <https://arxiv.org/abs/2606.03363>.
- Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. arXiv preprint arXiv:2304.11477, 2023. URL <https://arxiv.org/abs/2304.11477>.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl\_a\_00638. URL <https://arxiv.org/abs/2307.03172>. arXiv:2307.03172.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, et al. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations (ICLR)*, 2024b. URL <https://arxiv.org/abs/2308.03688>. arXiv:2308.03688.
- Xiaozhe Liu, Ruowang Zhang, Amir H. Abdi, Michel Galley, Zhikai Chen, Siheng Xiong, Xiaoqian Wang, and Jing Gao. Do Proactive Agents Need an LLM to Decide When to Act? arXiv preprint arXiv:2605.30152, 2026. URL <https://arxiv.org/abs/2605.30152>.
- Andy Lo, Albert Q. Jiang, Wenda Li, and Mateja Jamnik. End-to-End Ontology Learning with Large Language Models. In *Advances in Neural Information Processing Systems 37*, pp. 87184–87225, 2024. doi: 10.52202/079017-2767. URL <http://www.proceedings.com/079017-2767.html>.
- Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, Weiwen Liu, Yasheng Wang, Zhiyuan Liu, Fangming Liu, and Maosong Sun. Proactive Agent: Shifting LLM Agents from Reactive Responses to Active Assistance. In *International Conference on Learning Representations*, pp. 47431–47457, 2025. URL [https://proceedings.iclr.cc/paper\\_files/paper/2025/hash/75c37811e830bf029584b1c6fac17726-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2025/hash/75c37811e830bf029584b1c6fac17726-Abstract-Conference.html).
- Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *International Conference on Learning Representations*, 2024. URL <https://arxiv.org/abs/2310.01061>.
- Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. MCP-Universe: Benchmarking Large Language Models with Real-World Model Context Protocol Servers. arXiv preprint arXiv:2508.14704, 2025. URL <https://arxiv.org/abs/2508.14704>.
- Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. In *International Conference on Learning Representations*, pp. 52782–52806, 2025. URL [https://proceedings.iclr.cc/paper\\_files/paper/2025/hash/830b1abc6d2da85f23d41169fa44d185-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2025/hash/830b1abc6d2da85f23d41169fa44d185-Abstract-Conference.html).
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747. URL <https://aclanthology.org/2024.acl-long.747>.

- Costas Mavromatis and George Karypis. GNN-RAG: Graph Neural Retrieval for Efficient Large Language Model Reasoning on Knowledge Graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16682–16699, 2025. doi: 10.18653/v1/2025.findings-acl.856. URL <https://aclanthology.org/2025.findings-acl.856>.
- Dennis McCarthy and Umeshwar Dayal. The architecture of an active database management system. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, pp. 215–224, 1989. doi: 10.1145/67544.66946. URL <https://doi.org/10.1145/67544.66946>.
- Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for General AI Assistants. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://arxiv.org/abs/2311.12983>. arXiv:2311.12983.
- Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text. In *The Semantic Web – ISWC 2023*, pp. 247–265, 2023. doi: 10.1007/978-3-031-47243-5\_14. URL [https://link.springer.com/10.1007/978-3-031-47243-5\\_14](https://link.springer.com/10.1007/978-3-031-47243-5_14).
- Belinda Mo, Kyssen Yu, Joshua Kazdan, Proud Mpala, Lisa Yu, Charilaos Kanatsoulis, and Sanmi Koyejo. KGGen: Extracting Knowledge Graphs from Plain Text with Language Models. In *Advances in Neural Information Processing Systems 38*, pp. 33962–33985, 2025. doi: 10.52202/085713-1010. URL <http://www.proceedings.com/085713-1010.html>.
- Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. NoLiMa: Long-Context Evaluation Beyond Literal Matching. In *International Conference on Machine Learning*, 2025. URL <https://arxiv.org/abs/2502.05167>.
- Zahra Moslemi, Keerthi Koneru, Yen-Ting Lee, Sheethal Kumar, and Ramesh Radhakrishnan. POLARIS: Typed Planning and Governed Execution for Agentic AI in Back-Office Automation. arXiv preprint arXiv:2601.11816, 2026. URL <https://arxiv.org/abs/2601.11816>.
- Anil Nigam and Nathan S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428–445, 2003. doi: 10.1147/SJ.423.0428. URL <https://doi.org/10.1147/SJ.423.0428>.
- Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale knowledge graphs. *Communications of the ACM*, 62(8):36–43, 2019. doi: 10.1145/3331166. URL <https://dl.acm.org/doi/10.1145/3331166>.
- Abdulsobur Oyewale and Tommaso Soru. LLM-Driven Ontology Construction for Enterprise Knowledge Graphs. In *2026 International Conference on Semantic Computing (ICSC)*, pp. 267–270, 2026. doi: 10.1109/icsc67292.2026.00044. URL <https://ieeexplore.ieee.org/document/11486381/>.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. URL <https://arxiv.org/abs/2310.08560>.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599, 2024. doi: 10.1109/tkde.2024.3352100. URL <https://ieeexplore.ieee.org/document/10387715/>.
- 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. In *ACM Symposium on User Interface Software and Technology (UIST)*, 2023. URL <https://arxiv.org/abs/2304.03442>. arXiv:2304.03442.
- Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large Language Model Connected with Massive APIs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL <https://arxiv.org/abs/2305.15334>. arXiv:2305.15334.

- Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph Retrieval-Augmented Generation: A Survey. *ACM Transactions on Information Systems*, 44(2):1–52, 2025. doi: 10.1145/3777378. URL <https://dl.acm.org/doi/10.1145/3777378>.
- Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking Data to Ontologies. In *Journal on Data Semantics X*, pp. 133–173. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-77688-8\_5. URL [http://link.springer.com/10.1007/978-3-540-77688-8\\_5](http://link.springer.com/10.1007/978-3-540-77688-8_5).
- Mohammadreza Pourreza and Davood Rafiei. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction. In *Advances in Neural Information Processing Systems 36*, pp. 36339–36348, 2023. doi: 10.52202/075280-1577. URL <http://www.proceedings.com/075280-1577.html>.
- María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. OOPS! (Ontology Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems*, 10(2):7–34, 2014. doi: 10.4018/ijswis.2014040102. URL <https://doi.org/10.4018/ijswis.2014040102>.
- Cheng Qian, Chi Han, Yi R. Fung, Yujia Qin, Zhiyuan Liu, and Heng Ji. CREATOR: Tool Creation for Disentangling Abstract and Concrete Reasoning of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023. arXiv:2305.14318.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://arxiv.org/abs/2307.16789>. arXiv:2307.16789.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv preprint arXiv:2501.13956, 2025. URL <https://arxiv.org/abs/2501.13956>.
- Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 431–445, 2023. doi: 10.18653/v1/2023.emnlp-demo.40. URL <https://aclanthology.org/2023.emnlp-demo.40/>.
- Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, 2009. doi: 10.1561/15000000019. URL <https://www.emerald.com/ftinr/article/4/1-2/1/1326508/The-Probabilistic-Relevance-Framework-BM25-and>.
- Michael Rumiantsev and Ivan Fokeev. Semantic Layers for Reliable LLM-Powered Data Analytics: A Paired Benchmark of Accuracy and Hallucination Across Three Frontier Models. arXiv preprint arXiv:2604.25149, 2026. URL <https://arxiv.org/abs/2604.25149>.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *International Conference on Learning Representations*, pp. 32628–32649, 2024. URL [https://proceedings.iclr.cc/paper\\_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html).
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL <https://arxiv.org/abs/2302.04761>. arXiv:2302.04761.
- Juan Sequeda, Dean Allemang, and Bryon Jacob. A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model’s Accuracy for Question Answering on Enterprise SQL Databases. In *Proceedings of the 7th Joint Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, pp. 1–12, 2024. doi: 10.1145/3661304.3661901. URL <https://dl.acm.org/doi/10.1145/3661304.3661901>.

- Kartik Sharma, Peeyush Kumar, and Yuning Li. OG-RAG: Ontology-grounded retrieval-augmented generation for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32950–32969, 2025. doi: 10.18653/v1/2025.emnlp-main.1674. URL <https://aclanthology.org/2025.emnlp-main.1674>.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-0377. URL <https://arxiv.org/abs/2303.11366>. arXiv:2303.11366.
- Gundeep Singh, Parsa Kavehzadeh, Jing Xia, Xue-Yong Fu, Julien Bouvier Tremblay, Md Tahmid Rahman Laskar, Vincent Lum, and Shashi Bhushan TN. Beyond Text-to-SQL: An Agentic LLM System for Governed Enterprise Analytics APIs. arXiv preprint arXiv:2605.21027, 2026. URL <https://arxiv.org/abs/2605.21027>.
- Olly Styles, Sam Miller, Patricio Cerda-Mardini, Tanaya Guha, Victor Sanchez, and Bertie Vidgen. WorkBench: a Benchmark Dataset for Agents in a Realistic Workplace Setting. arXiv preprint arXiv:2405.00823, 2024. URL <https://arxiv.org/abs/2405.00823>.
- 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>. arXiv:2309.02427.
- Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *International Conference on Learning Representations*, 2024. URL <https://arxiv.org/abs/2307.07697>.
- Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. CHES: Contextual Harnessing for Efficient SQL Synthesis. arXiv preprint arXiv:2405.16755, 2024. URL <https://arxiv.org/abs/2405.16755>.
- Hao Tang, Darren Key, and Kevin Ellis. WorldCoder, a Model-Based LLM Agent: Building World Models by Writing Code and Interacting with the Environment. In *Advances in Neural Information Processing Systems 37*, pp. 70148–70212, 2024. doi: 10.52202/079017-2243. URL <http://www.proceedings.com/079017-2243.html>.
- Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL <https://arxiv.org/abs/2407.18901>. arXiv:2407.18901.
- Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, LinZheng Chai, Zhao Yan, Qian-Wen Zhang, Di Yin, Xing Sun, and Zhoujun Li. MAC-SQL: A Multi-Agent Collaborative Framework for Text-to-SQL. In *Proceedings of the International Conference on Computational Linguistics*, 2025a. URL <https://arxiv.org/abs/2312.11242>.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlikar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024a. URL <https://arxiv.org/abs/2305.16291>. arXiv:2305.16291.
- Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents. In *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering*, pp. 2938–2950, 2026a. doi: 10.1145/3744916.3764546. URL <https://doi.org/10.1145/3744916.3764546>.
- Ningning Wang, Xavier Hu, Pai Liu, He Zhu, Yue Hou, Heyuan Huang, Shengyu Zhang, Jian Yang, Jiaheng Liu, Ge Zhang, Changwang Zhang, Jun Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Efficient Agents: Building Effective Agents While Reducing Cost. arXiv preprint arXiv:2508.02694, 2025b. URL <https://arxiv.org/abs/2508.02694>.

- Ruiyang Wang, Hao-Lun Hsu, Swarajh Mehta, Jiwoo Kim, Zhihao Dou, and Miroslav Pajic. GAVEL: Graph World Models for Verified and Efficient Long-Horizon LLM Task Planning. arXiv preprint arXiv:2609.19315, 2026b. URL <https://arxiv.org/abs/2609.19315>.
- Weixuan Wang, Dongge Han, Daniel Madrigal Diaz, Jin Xu, Victor Rühle, and Saravan Rajmohan. OdysseyBench: Evaluating LLM Agents on Long-Horizon Complex Office Application Workflows. arXiv preprint arXiv:2508.09124, 2025c. URL <https://arxiv.org/abs/2508.09124>.
- Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable Code Actions Elicit Better LLM Agents. In *International Conference on Machine Learning (ICML)*, 2024b. URL <https://arxiv.org/abs/2402.01030>. arXiv:2402.01030.
- Yilun Wang, Guangba Yu, Haiyu Huang, Yujie Huang, Zirui Wang, Pengfei Chen, and Michael R. Lyu. Cloud-OpsBench: A Reproducible Benchmark for Agentic Root Cause Analysis in Cloud Systems. arXiv preprint arXiv:2603.00468, 2026c. URL <https://arxiv.org/abs/2603.00468>.
- Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Parag Shah, Yujia Bao, and Eugene Siow. MCP-Bench: Benchmarking Tool-Using LLM Agents with Complex Real-World Tasks via MCP Servers. In *International Conference on Learning Representations*, pp. 97377–97425, 2026d. URL [https://proceedings.iclr.cc/paper\\_files/paper/2026/hash/9e4b14eb6f16fe7b5818a8d633a0606a-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2026/hash/9e4b14eb6f16fe7b5818a8d633a0606a-Abstract-Conference.html).
- Zhiruo Wang, Graham Neubig, and Daniel Fried. TroVE: Inducing Verifiable and Efficient Toolboxes for Solving Programmatic Tasks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 51177–51191, 2024c. URL <https://proceedings.mlr.press/v235/wang24az.html>.
- Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. OfficeBench: Benchmarking Language Agents across Multiple Applications for Office Automation. arXiv preprint arXiv:2407.19056, 2024d. URL <https://arxiv.org/abs/2407.19056>.
- Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding. In *International Conference on Learning Representations*, 2024e. URL <https://arxiv.org/abs/2401.04398>.
- Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. In *International Conference on Machine Learning (ICML)*, 2025d. URL <https://arxiv.org/abs/2409.07429>. arXiv:2409.07429.
- Jennifer Widom and Stefano Ceri (eds.). *Active Database Systems: Triggers and Rules for Advanced Database Processing*. Morgan Kaufmann, 1995. URL <https://shop.elsevier.com/books/active-database-systems/widom/978-0-08-049856-0>.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2410.10813>.
- Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM Agents via Knowledge-Enabled Reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68316–68342. PMLR, 2025. URL <https://proceedings.mlr.press/v267/xiang25a.html>.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649, 2024. doi: 10.1145/3626772.3657878. URL <https://dl.acm.org/doi/10.1145/3626772.3657878>.

- Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, et al. TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. URL <https://arxiv.org/abs/2412.14161>. arXiv:2412.14161.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110*, 2025b. URL <https://arxiv.org/abs/2502.12110>.
- Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Ion Stoica, Joseph E. Gonzalez, Tianjun Zhang, and Shishir G. Patil. Berkeley Function-Calling Leaderboard. UC Berkeley Gorilla project, 2024. URL [https://gorilla.cs.berkeley.edu/blogs/8\\_berkeley\\_function\\_calling\\_leaderboard.html](https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html). Accessed 2026-10-08.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*, 2023. URL <https://arxiv.org/abs/2210.03629>. arXiv:2210.03629.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan.  $\tau$ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *International Conference on Learning Representations*, pp. 9965–10017, 2025. URL [https://proceedings.iclr.cc/paper\\_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html](https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html).
- Yining Ye, Xin Cong, Shizuo Tian, Jiannan Cao, Hao Wang, Yujia Qin, Yaxi Lu, Heyang Yu, Huadong Wang, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ProAgent: From Robotic Process Automation to Agentic Process Automation. *arXiv preprint arXiv:2311.10751*, 2023. URL <https://arxiv.org/abs/2311.10751>.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, 2018. doi: 10.18653/v1/d18-1425. URL <http://aclweb.org/anthology/D18-1425>.
- Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R. Fung, Hao Peng, and Heng Ji. CRAFT: Customizing LLMs by Creating and Retrieving from Specialized Toolsets. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.17428.
- Mario Zechner. pi: a minimal agent harness and coding agent. <https://github.com/badlogic/pi-mono>, 2025.
- Zhen Zeng, William Watson, Nicole Cho, Saba Rahimi, Shayleen Reynolds, Tucker Balch, and Manuela Veloso. FlowMind: Automatic Workflow Generation with LLMs. In *4th ACM International Conference on AI in Finance*, pp. 73–81, 2023. doi: 10.1145/3604237.3626908. URL <https://dl.acm.org/doi/10.1145/3604237.3626908>.
- Bohui Zhang, Valentina Anita Carriero, Katrin Schreiberhuber, Stefani Tsaneva, Lucía Sánchez González, Jongmo Kim, and Jacopo de Berardinis. OntoChat: A Framework for Conversational Ontology Engineering Using Language Models. In *The Semantic Web: ESWC 2024 Satellite Events*, pp. 102–121, 2025a. doi: 10.1007/978-3-031-78952-6\_10. URL [https://link.springer.com/10.1007/978-3-031-78952-6\\_10](https://link.springer.com/10.1007/978-3-031-78952-6_10).
- Bowen Zhang and Harold Soh. Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9820–9836, 2024. doi: 10.18653/v1/2024.emnlp-main.548. URL <https://aclanthology.org/2024.emnlp-main.548>.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, et al. AFlow: Automating Agentic Workflow Generation. In *International Conference on Learning Representations (ICLR)*, 2025b. URL <https://arxiv.org/abs/2410.10762>. arXiv:2410.10762.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamamuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, et al. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. *arXiv preprint arXiv:2510.04618*, 2025c.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A Survey on the Memory Mechanism of Large Language Model based Agents. *arXiv preprint arXiv:2404.13501*, 2024. URL <https://arxiv.org/abs/2404.13501>.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, and Yu Su. SkillWeaver: Web Agents can Self-Improve by Discovering and Honing Skills. *arXiv preprint arXiv:2504.07079*, 2025.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web*, 27(5), 2024. doi: 10.1007/s11280-024-01297-w. URL <https://link.springer.com/10.1007/s11280-024-01297-w>.

Naama Zwerdling, David Boaz, Ella Rabinovich, Guy Uziel, David Amid, and Ateret Anaby Tavor. Towards Enforcing Company Policy Adherence in Agentic Workflows. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 595–606, 2025. doi: 10.18653/v1/2025.emnlp-industry.41. URL <https://aclanthology.org/2025.emnlp-industry.41/>.

## A 证明

**命题 1.** 运行框架在每次模型调用时重新发送对话，因此一次调用的输入包括系统文本、准备阶段的对话记录、先前请求及其回复与工具输出，以及当前请求及其至此产生的工具输出；这就是所述的和。图仅通过智能体运行的命令进入输入：命令输出属于  $o_k$ （请求结束后属于  $h_i$ ），研究或构建图属于  $p$ 。以视图替代原始行会改变工具输出大小，以及包含这些输出的历史大小，其他部分不变；若无图的智能体已能编写返回视图的查询（聚合、过滤后的行），两者的工具输出项具有相同量级。对于长上下文策略，考虑参与聚合的  $n$  行组成的集合  $X \subseteq D$  上的聚合  $\sum_{x \in X} f(x)$ （例如 所有未结订单的金额）。任何依据上下文作答的策略都必须将每个  $x \in X$  放入上下文，因为改变一个被遗漏的  $x$  会改变真实答案，却不改变上下文；因此，对于这类聚合，上下文包含  $\Omega(n)$  个 token。对于检索器，若对聚合有贡献的  $m$  个条目存于  $m$  个不同文本块，且检索器返回  $r < m$  个文本块，则上下文中至少缺少一个有贡献的条目；同一论证表明，答案不可能对所有实例都精确。□

**命题 2.** 引擎仅在  $c_a(\theta, D)$  成立时提交  $a$  的应用，且  $c_a \Rightarrow \pi$ ，因此每次已提交应用均满足  $\pi$ 。对于上下文内策略， $N$  次写入各以概率  $\varepsilon$  违反  $\pi$ ，且相互独立，因此没有违反制度的操作的概率为  $(1 - \varepsilon)^N$ 。若  $c_a$  在集合  $W$  上有误，即接受某些满足  $\neg \pi(\theta, D)$  的  $(\theta, D) \in W$ ，或拒绝某些满足  $\pi(\theta, D)$  的输入，则引擎对  $W$  中每个输入都作出同一个确定性的错误决策，因此  $W$  中各次写入的错误完全相关。□

**命题 3.** 自动化服务在每个变更集上对  $\kappa_\gamma$  求值，并在其成立时应用  $\epsilon_\gamma$ ；模型不参与，因此以模型调用计的处理成本为零，正确的条件和效果能处理每个触发事件，后续请求看到的事件已被处理。仅从摘要获知事件的智能体，必须在读取摘要后于后续请求的一次模型调用中采取行动；只有模型识别出触发事件时才会处理它，其概率为  $1 - \varepsilon_e$ 。□

**命题 4.**  $n$  个请求后的累计成本，没有图时为  $M_0 + nc_0$ ，有图时为  $M_G + nc_G$ ；两者在  $n^* = (M_G - M_0)/(c_0 - c_G)$  时相等，若  $c_0 > c_G$  且  $M_G > M_0$ ，该值为正且有限；若  $c_G \geq c_0$  且  $M_G > M_0$ ，有图时的累计成本始终更高。□

B 带类型的图接口

表 3 列出图接口的命令。输出由主机渲染为对齐文本（默认至多 50 行）或 JSON；错误消息指出出错项，并建议一种修复方式。过滤条件使用 Mongo 风格 JSON（\$eq、\$gt、\$in、\$contains、\$and、\$or、...），随后转换为内核的对象集过滤条件；聚合和遍历在引擎中运行。在建模模式下，图最初只有一个平台类型（ErpRequest，即发件箱），帮助文本提供六步工作流程、每种定义的模板，以及一个基于两张库存表、不编码任何制度的可运行示例。

表 3: 带类型图接口的命令及开放这些命令的模式（L1 为只读图，L2 加入动作，L3 加入函数与自动化规则，SELF 加入建模）。

| 分组    | 命令                                                                                                                                                                                | 模式           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 读取    | types, describe T, objects T --where --select --order-by, object T pk, links T pk L, traverse T --via L1, L2 --target-where, count T, aggregate T --metrics --group-by, history T | 全部           |
| 写入    | actions, action A, apply A --param k=v [--validate-only], tx-catalog                                                                                                              | L2, L3, SELF |
| 计算与响应 | functions, call F, automation list show create update delete runs                                                                                                                 | L3, SELF     |
| 建模    | template K, define FILE, map FILE, function publish FILE, validate, publish, model export                                                                                         | SELF         |

C 系统细节

主机。 每种对象类型由 ERP 表或只读查询支撑，并通过值变换处理编码。每批事件和每次写入后，主机读取 ERP 变更日志，将变更作为数据源编辑应用于镜像对象，并让自动化服务对变更集求值；在智能体通过原始事务写入的只读模式下，主机还在每次读取前拉取日志。动作的规则或函数生成 ErpRequest 对象；主机以动作名义按顺序将其作为 ERP 事务执行，因此失败事务会使动作应用失败并返回 ERP 的消息，随后同步结果。这也是生产部署的写入路径。能力模式分别开放仅读取、读取与动作、除建模外的全部能力，或全部能力。

沙箱与运行时池。 每次执行以非特权用户身份在临时系统单元中运行。它仅能看到语言运行时、两个命令行客户端及自身工作区（看不到基准、参考判定器、参考图、其他会话或其他租户），除通向代理与两个接口的本地套接字外，无法访问网络，且内存和进程数受限。所有研究进程共享一个限额为 2.5 GB 和三个核心的 cgroup，监控程序在池的可用内存低于 5 GB 时暂停研究。每次执行看到的工作区路径相同，因此同一会话的连续请求共享提示前缀，提供商的前缀缓存能够生效（附录 G）。长上下文和检索基线通过同一代理调用模型。

D 准备指令

每个智能体条件收到三种准备请求之一，均以“准备好后回复 READY”结束（SGM-self 为“模型发布后”）。

原始智能体。 工作周即将开始。为这项工作做好准备；不要更改 ERP。(1) 研究 ERP (erp tables、erp schema <TABLE>、erp sql) 和 docs/ 中的文档：每张表和每列的含义、编码、表的连接方式，以及每条制度规则。(2) 将此后需要的内容写入工作区，使请求处理快速且正确，例如 NOTES.md（含义、编码、连接、每项制度的检查清单），以及用于预计会重复进行的计算的辅助脚本（订单金额、可用信用额度、可用库存、补货需求、仓库分配）。(3) 用 erp sql 对照几行数据检查脚本。

图条件（KG、+Actions、SGM-expert）。 工作周即将开始。为这项工作做好准备；不要更改 ERP。(1) 研究语义图 (sg help、sg types、sg describe <Type>，在具有动作的条件中还有 sg actions) 及 docs/ 中的制度。(2) 将此后需要的内容写入工作区，使请求处理快速且正确，例如在 NOTES.md 中列出处理每类请求会使用的类型、属性、链接（和动作），以及每项制度的检查清单。(3) 尝试几个查询，检查你是否正确读取了图。

SGM-self。 工作周即将开始。在此之前，构建你整个工作周会使用的语义图。好的模型会使后续请求处理快速且正确。不要更改 ERP。(1) 研究 ERP 及 docs/ 中的文档（数据字典、制度、事务参考）。(2) 为业务实体定义对象类型，使用清晰的名称和描述（含义、单位、编码值）。用 sg map 将每种类型映射到其表，将编码转换为有意义的值。定义类型间的链接。(3) 对制度及典型问题所需的派生量（例如金额、余额、可用量）建模，将其作为 SQL 支撑的映射中的属性，或定义为函数。(4) 为制度约束的变更定义动作。每个动作检查其制度（通过条件，或抛出错误的函数），并经发件箱发送所需的 ERP

事务。阅读 sg help action, 用 --validate-only 测试每个动作。(5) 运行 sg validate, 再运行 sg publish。用 erp sql 对照几个对象和计数。(6) 编写简短的 NOTES.md, 描述你的模型。

SGM-self 的指令比其他条件更具指导性, 接口帮助还提供模板及基于两张库存表的可运行示例: 这种不对称有利于 SGM-self 的建模, 也是第 6.3 节测量的一部分。

E 专家图

专家图包含 13 种镜像对象类型: Customer、Material、Vendor、Warehouse、InventoryBalance、SalesOrder、SalesOrderLine、Shipment、Invoice、PurchaseOrder、PurchaseOrderLine、PriceChange 和 Notification; 另有一个保存审批阈值的 PolicySetting 类型, 因此第四天的变更只需一个动作。编码转换为标签 (客户分层 A → 金牌, 状态 H → 冻结)。八种类型由只读查询支撑, 并加入派生量: 折扣后订单金额、未结状态、客户的未结订单金额与可用信用额度、客户是否为战略客户及其最高折扣、物料的可用库存、未结采购订单与补货需求, 以及发票逾期天数。十二种链接类型连接客户、订单、明细、物料、供应商、库存、仓库、发票、发运和采购订单。十一个动作通过函数规则实现制度: 录入销售订单 (信用、折扣、审批及优先级规则)、修改订单明细 (包括重新检查信用)、取消、审批、分配 (仓库选择)、检查新 EDI 订单、物料再订货 (幂等)、变更标价、冻结客户信用、通知, 以及修改制度设置。四个查询函数计算订单金额、客户可用信用额度、订单的最佳仓库, 以及物料的补货需求。仅使用该图及三条自动化规则脚本化智能体能够解决参考会话中的每个请求, 并处理每个持续规则触发事件 (第 5 节)。

F 补充结果

表 4: 输入的来源 (种子数同表 2)。准备阶段上下文: 首个请求的对话记录字符数, 此后每次调用均重新发送; 工具输出: 每请求命令返回的字符数。

| 条件           | 令牌/调用 (k) | 调用/请求 | 准备阶段上下文 (千字符) | 工具输出/请求 (字符) |
|--------------|-----------|-------|---------------|--------------|
| 智能体 (原始 ERP) | 88.1      | 3.62  | 149           | 1203         |
| 知识图谱 (L1)    | 90.8      | 3.97  | 120           | 2348         |
| + 动作 (L2)    | 76.2      | 3.57  | 134           | 1833         |
| SGM-专家图 (L3) | 99.2      | 3.69  | 147           | 2285         |
| SGM-自建       | 159.6     | 3.21  | 454           | 1072         |

表 5: 按规则统计及时满足的持续规则义务, 由变更日志重新评分 (附录 G), 同时列出实时检查点值。SR1: 库存低于再订货点时补货; SR2: 新 EDI 订单到达时检查; SR3: 分配战略客户优先级为 1 的订单。由自动化规则完成: 已满足义务中由自动化规则满足的比例。

| 条件           | SR1 (%) | SR2 (%) | SR3 (%) | 召回率 (%) | 精确率 (%) | 自动化完成 (%) | 原判召回率 (%) | 原判精确率 (%) |
|--------------|---------|---------|---------|---------|---------|-----------|-----------|-----------|
| 长上下文 LLM     | 77.8    | 71.4    | 20.0    | 63.3    | 72.4    | 0.0       | 58.9      | 60.0      |
| RAG          | 0.0     | 5.9     | 0.0     | 4.6     | 75.0    | 0.0       | 4.3       | 75.0      |
| 智能体 (原始 ERP) | 90.0    | 100.0   | 85.3    | 96.4    | 98.6    | 0.0       | 96.3      | 80.7      |
| 知识图谱 (L1)    | 100.0   | 99.6    | 100.0   | 99.7    | 98.9    | 0.0       | 99.2      | 73.0      |
| + 动作 (L2)    | 96.7    | 100.0   | 49.2    | 90.5    | 98.4    | 0.0       | 89.0      | 82.0      |
| SGM-专家图 (L3) | 100.0   | 100.0   | 55.7    | 92.6    | 99.5    | 80.0      | 91.1      | 81.6      |
| SGM-自建       | 100.0   | 100.0   | 74.8    | 95.7    | 99.7    | 82.2      | 94.7      | 80.5      |

整个研究, 包括试运行、中止的首次主实验、清空对话变体、规模研究和迁移研究, 共进行了 9,167 次模型调用, 标价成本为 155.94 CNY; 大多数运行处于提供商的半价时段, 实付金额为 89.68 CNY (试运行 23.42、中止运行 8.84、主研究 41.25、清空对话变体 9.24、规模研究 4.94、迁移研究 1.98 CNY)。

表 6: 含 40 个请求的会话中的请求类别。

| 类别        | 示例                      | # | 判分依据         |
|-----------|-------------------------|---|--------------|
| F1 查找     | 指定客户的付款条件与信用额度          | 4 | 答案           |
| F2 多跳     | 客户最近一笔未结订单所含物料的供应商      | 5 | 答案 (集合)      |
| F3 聚合     | 按地区统计金牌客户的未结订单金额        | 6 | 答案 (容差 0.5%) |
| F4 条件分析   | 有逾期 >30 天发票且有未结订单的客户    | 4 | 答案 (集合)      |
| F5 制度约束写入 | 录入要求 12% 折扣的电话订单        | 8 | ERP 差异 + 答案  |
| F6 批量更新   | 供应商涨价 4%: 更新其在用物料       | 3 | ERP 差异       |
| F7 持续规则   | 从现在起, 每笔新 EDI 订单到达时都要检查 | 4 | 后续事件         |
| F8 连续性    | 列出你今天录入的订单              | 2 | 答案           |
| F9 报告     | 包含五项数值的日终报告             | 4 | 答案 (逐字段)     |

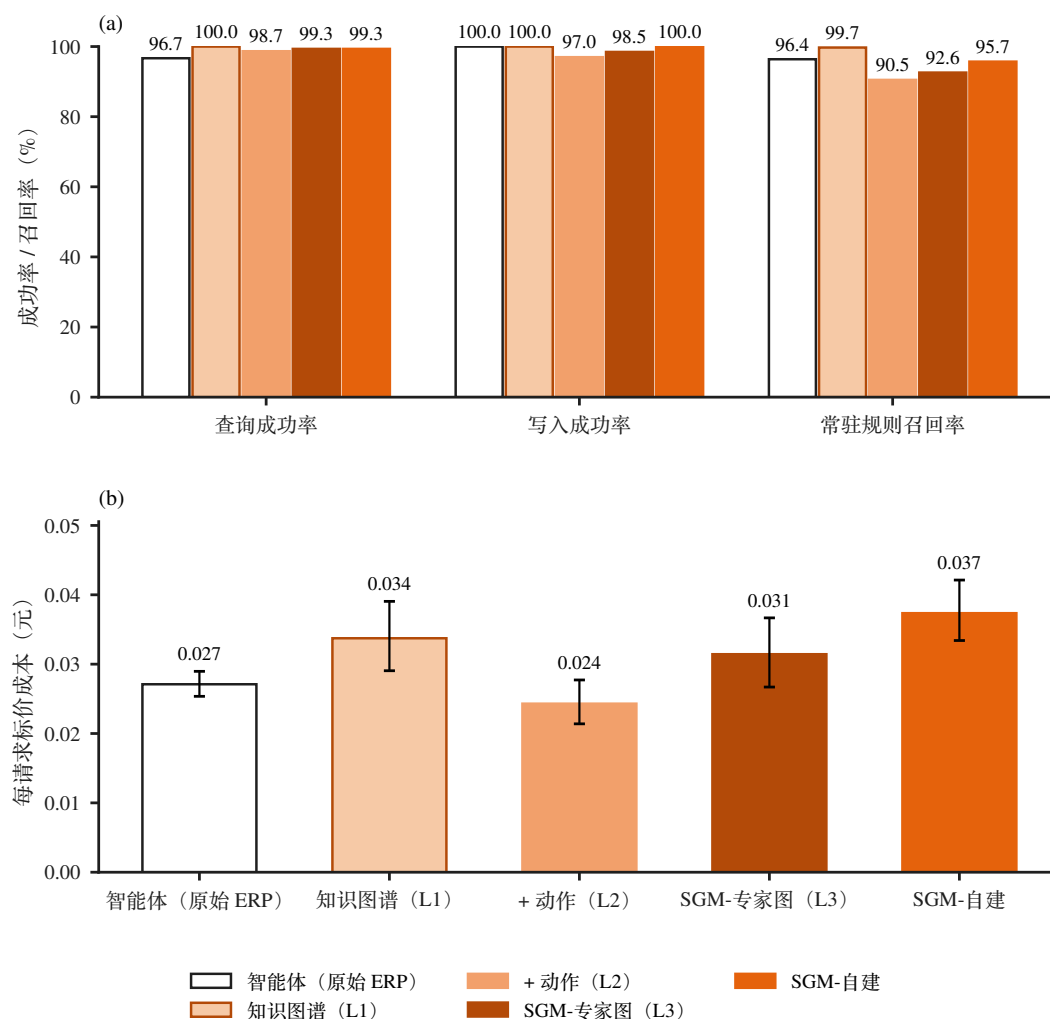


图 2: M 规模的消融阶梯 (附 SGM-self 作对照): (a) 查询与写入成功率, 以及及时满足的持续规则义务比例; (b) 每请求的标价成本及其 95% 自助法区间。种子数同表 2。

表 7: 按类别统计的请求成功率（%，M 规模；F7 持续指令通过规则检查点判分）。

| 条件          | F1    | F2    | F3    | F4    | F5    | F6    | F7 | F8    | F9    |
|-------------|-------|-------|-------|-------|-------|-------|----|-------|-------|
| 长上下文 LLM    | 100.0 | 80.0  | 50.0  | 50.0  | 100.0 | 33.3  | –  | 100.0 | 100.0 |
| RAG         | 66.7  | 0.0   | 16.7  | 0.0   | 37.5  | 0.0   | –  | 50.0  | 25.0  |
| 智能体（原始 ERP） | 100.0 | 100.0 | 94.4  | 95.8  | 100.0 | 100.0 | –  | 100.0 | 91.7  |
| 知识图谱（L1）    | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | –  | 100.0 | 100.0 |
| + 动作（L2）    | 100.0 | 100.0 | 94.4  | 100.0 | 100.0 | 88.9  | –  | 100.0 | 100.0 |
| SGM-专家图（L3） | 100.0 | 100.0 | 100.0 | 95.8  | 97.9  | 100.0 | –  | 100.0 | 100.0 |
| SGM-自建      | 100.0 | 100.0 | 100.0 | 95.8  | 100.0 | 100.0 | –  | 100.0 | 100.0 |

表 8: SGM-self 构建的图（会话均值）。覆盖率和一致性以专家图为参照，专家图只是多种有效设计之一；原始 SQL 请求：智能体至少使用一次 SQL 的请求。

| 指标             | SGM-self（均值） |
|----------------|--------------|
| 构建类型数          | 14.7         |
| 构建链接数          | 13.8         |
| 构建动作数          | 15.3         |
| 构建函数数          | 18.2         |
| 构建自动化数         | 4.2          |
| 类型覆盖率（%）       | 100.0        |
| 属性映射一致率（%）     | 48.3         |
| 链接召回率（%）       | 60.6         |
| 策略覆盖率（%）       | –            |
| 自动化正确率（%）      | –            |
| 原始 SQL 请求占比（%） | 86.0         |

表 9: 探索性变体：准备请求后清空对话（cleared），与相同种子上的主实验对比。

| 条件                | 种子数 | 成功率（%） | 令牌/请求 (k) | 令牌/调用 (k) | 元/请求  | 耗时中位数 (s) | 准备成本（元） |
|-------------------|-----|--------|-----------|-----------|-------|-----------|---------|
| 智能体（原始 ERP）       | 3   | 96.3   | 298.7     | 82.3      | 0.026 | 10.9      | 0.22    |
| 智能体（原始 ERP）（清空对话） | 3   | 97.2   | 196.0     | 55.5      | 0.022 | 9.9       | 0.43    |
| SGM-专家图（L3）       | 3   | 99.1   | 347.1     | 94.4      | 0.031 | 9.6       | 0.15    |
| SGM-专家图（L3）（清空对话） | 3   | 99.1   | 284.2     | 71.4      | 0.029 | 8.9       | 0.24    |
| SGM-自建            | 3   | 100.0  | 467.3     | 154.3     | 0.032 | 8.9       | 0.83    |
| SGM-自建（清空对话）      | 3   | 100.0  | 248.7     | 66.8      | 0.025 | 8.7       | 1.36    |

表 10: 按阶段核算报告运行中的模型调用，分别按标价和实付价格计算（提供商非高峰时段为半价）。

| 阶段       | 会话数 | 请求数  | 调用数  | 准备成本（元） | 标价成本（元） | 实付成本（元） |
|----------|-----|------|------|---------|---------|---------|
| main     | 37  | 1476 | 6279 | 14.07   | 95.04   | 47.52   |
| pilot    | 5   | 69   | 253  | 0.53    | 9.41    | 9.41    |
| scaling  | 7   | 140  | 716  | 1.74    | 9.89    | 4.94    |
| transfer | 2   | 120  | 437  | 0.22    | 3.95    | 1.98    |

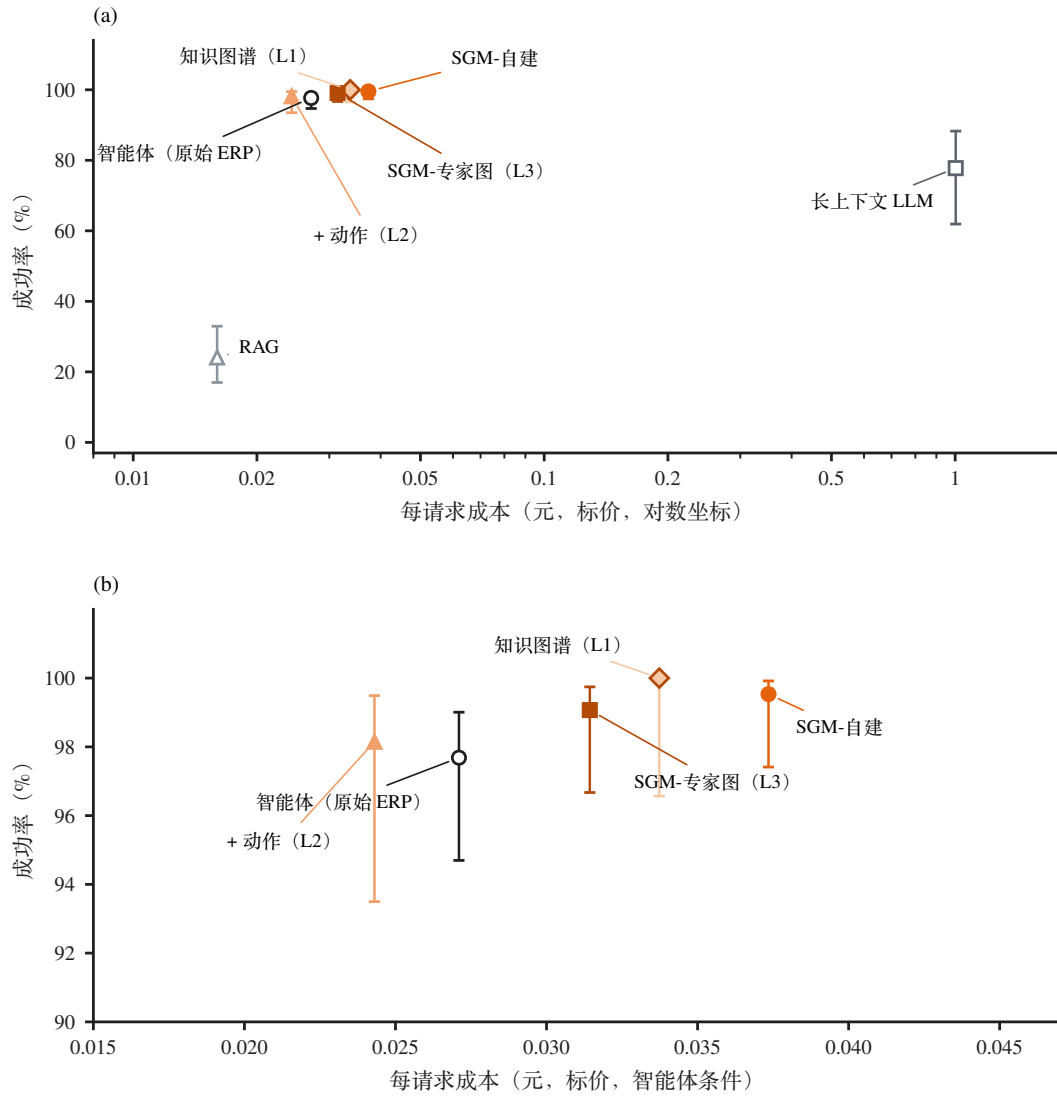


图 3: M 规模的成功率与每请求标价成本: (a) 全部条件 (对数尺度); (b) 智能体条件。误差条为基于请求自助法的 95% 区间; 实心标记为图条件, 空心标记为基线。

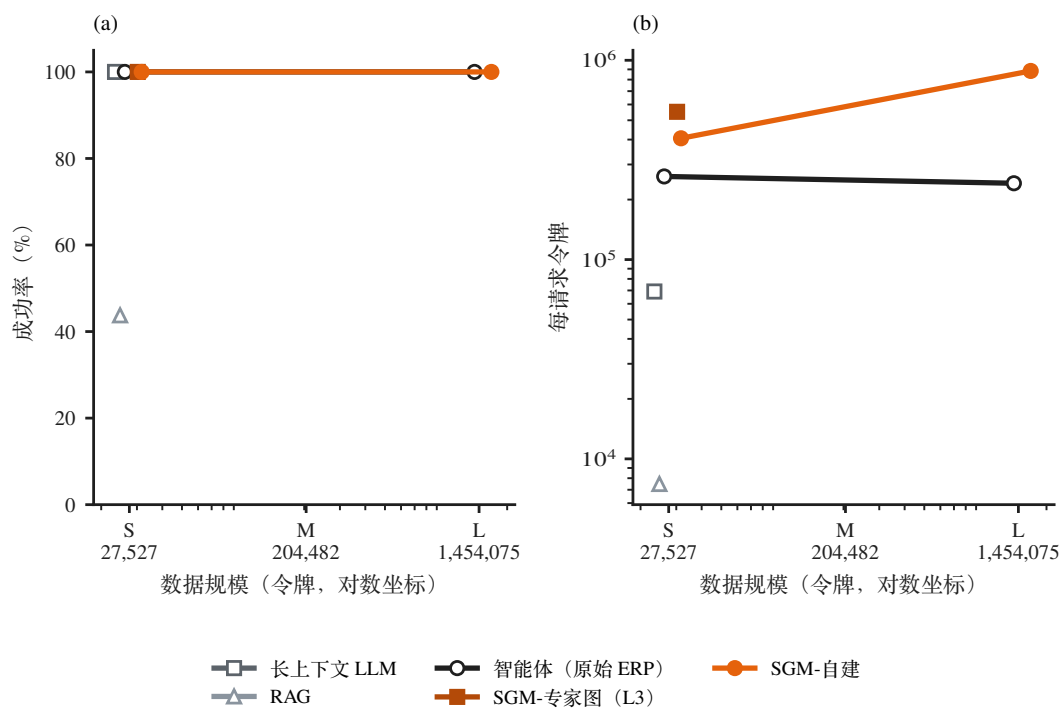


图 4: 规模变化: S 和 L 规模下, 一个会话前 20 个请求的成功率和每请求 token 数 (数据大小使用对数轴; M 规模的主研究请求组合不同, 见表 2)。SGM-expert 只在 S 规模运行; L 规模只运行了原始智能体和 SGM-self: 数据超过长上下文提示的上下文窗口, 而检索在 M 规模已是下限基线。

## G 评估规范

我们报告所有影响运行或判分的问题，因为它们关系到任何允许智能体使用 shell 的基准。第 (i) 至 (iii) 项在试运行中暴露，并在主研究前修复；第 (iv) 至 (x) 项在主研究期间或之后出现。(i) 参考实现泄露。最初的沙箱以只读方式暴露了研究环境的安装目录。一个 SGM-self 智能体在检查其命令行客户端的包装脚本时，发现了参考图的源文件，并将其函数的副本作为自己的函数发布。最终的沙箱仅向每次执行暴露语言运行时、两个客户端及其自身的工作区。(ii) 指令泄露。最初的制度文档重述了持续指令，因此一个具备自动化规则能力的智能体在收到这些指令之前就配置了相应规则。现在，持续指令仅出现在下达它们的请求中。(iii) 评分。对于智能体忽略的规则触发事件，此前每个后续检查点都会重复计数；在另一请求的时间窗口内产生的规则效果，则被计为该请求的范围外写入。现在，每个触发事件只计数一次，规则相关工作仅由规则检查点评分，且指令从下达时起生效，无论智能体如何回复。试运行数据仅用于成本估算，不纳入任何报告结果。(iv) 提示前缀稳定性。首次主实验将每次执行的工作空间暴露在新的主机路径下。智能体运行框架将工作目录写入系统提示，导致每个请求的提示前缀都变化：提供商的前缀缓存在每请求的第一次模型调用时未命中，智能体还尝试进入先前请求中已不存在的目录。直接调用模型的条件使用固定前缀，后接当前数据，不受影响，因此该实验伪差使智能体条件的成本和时间偏高。现在，与容器沙箱一样，每次执行中的工作空间都使用固定路径；主研究从头重新开始，首次运行的部分数据（71 个请求，发现伪差后停止）未使用。该次运行中，驱动程序还曾因缓存智能体流式输出而耗尽内存；现在沙箱内会丢弃流式更新。(v) 答案格式与最终状态写入。实时评分器将答案与参考判定器的原始值比较，因此将数字作为字符串给出，或给出编码文档含义的答案（订单状态 S 对应“Shipped”，付款条件 N30 对应“Net 30”）会被判错，尽管请求询问的是状态而非编码；图条件受影响最大，因为其映射将编码转换为含义。评分器还按每次中间变更判断写入，因此以一种状态创建、在同一请求内修正的电话订单会失败，自由文本字段中的备注或顺序不同的订单明细也会失败。我们用同一规则对每个条件重新评分：规范化数字和编码，接受编码或其文档含义，按请求窗口内每行的最终状态比较写入，忽略自由文本字段和明细顺序；预期值依据会话变更日志重建，为请求开始时的值。这改变了 71 个答案的评分（原始智能体为 15，SGM-self 为 10，SGM-expert 为 12）。重新评分前，成功率为 69.4% (LLM)、13.0% (RAG)、90.7% (Agent)、90.7% (KG)、89.8% (+Actions)、93.5% (SGM-expert) 和 94.9% (SGM-self)。(vi) 持续规则的时序。实时检查点直到规则下达后的第一批事件才登记规则义务。因此，若智能体在下达规则的请求中就处理现有积压，则被判为产生无触发依据的效果；忽略积压的智能体则被判为遗漏，两个错误对不同条件产生相反方向的影响。我们根据变更日志对每个会话重新评分：在初始世界上重放日志，每次变更后对规则触发条件求值，义务在触发条件成立的瞬间开始（对积压而言，是下达规则的请求开始时），并须在下一个请求结束前满足；效果若符合义务开始时或此后某个请求开始时状态上的制度决策，即满足该义务。表 5 并列报告实时值与重新评分值；正文使用后者。(vii) 只读图的陈旧读取。最初的 KG 会话所读取的图仅在事件批次和动作之后由主机同步；KG 通过原始事务写入，因此同一请求稍后的读取可能看不到智能体自己的写入。现在，该模式下主机在每次读取前拉取变更日志，KG 已在全部三个种子上从头重跑；最初的 KG 运行未使用。(viii) 时间限制。3 次执行超限（每请求 10 分钟，准备阶段 40 分钟）：一个 KG 会话的准备阶段，以及 SGM-expert 和 SGM-self 各一个请求；原始智能体没有超限。在我们运行期间检查的 SGM-self 请求中，智能体对沙箱的整个文件系统启动了递归文本搜索；另外两次执行在剩余时限内不再有模型调用，原因未知。第一版驱动程序在这种情况下会中止会话；我们改为将执行作为失败请求判分并保留其计量成本，然后继续，并从最后一个请求恢复受影响的会话。若没有这次超时，SGM-self 会在每个已判分请求上成功。在规模研究中，SGM-self 在 S 规模的准备阶段也超过了时限；随后它的提交因编排器 900 秒的租约过期而失败，会话恢复时把准备阶段记为超时（租约现为 3,000 秒）。(ix) 接口边界。沙箱隐藏研究安装目录和其他会话，客户端仅转发所属条件的命令；我们还审计了 37 个主实验会话的每条命令。没有智能体在其接口之外写入 ERP（不提供原始写入的条件中，原始写入为 0 次），也没有在不提供 SQL 的条件中运行 SQL（0 次）。智能体确实进行了探查：79 条命令读取了命令行客户端的源码（SGM-self 为 30 条，SGM-expert 为 21 条），SGM-self 在准备阶段发出了 17 条直接探查接口套接字的命令，均未写入 ERP。(x) 基础设施故障。在会话运行期间启动后续阶段时，共享的命令行客户端目录被替换，导致六次正在运行的执行丢失客户端，智能体因而搜索文件系统寻找客户端。排除规则依据时间固定，而非依据结果：若目录被替换时某次执行正在运行，则将其排除。我们停止了这些执行，将四个受影响的请求作为中断请求排除（主要条件中 2 个，清空对话变体中 2 个），为准备阶段受影响的两个清空对话会话分配新标识符并重新运行，同时修改启动程序，使其在原位置更新客户端。

## H 扩展相关工作

本节给出第2节所依据的完整综述。检索。2026年10月8日，我们在arXiv、Semantic Scholar和OpenAlex中检索了十个主题（基于图的推理与检索；长上下文与检索；本体、语义层及文本到SQL；知识图谱与本体构建；智能体记忆；长程与企业基准；工具与可执行世界模型；事件驱动智能体与主动规则；智能体成本；智能体构建的工具与技能），将4,239条记录合并为475个候选项，通过带类型的分类器及人工筛选标题与摘要，保留124项工作，并逐一依据arXiv、Crossref、DBLP或出版方核实。设计评审补充了24项针对性工作（运行时制度守卫、语义层、业务工件、视图更新、本体评估），以相同方式核实。

面向语言模型的知识图谱与语义层。知识图谱赋予实体和关系明确的类型与语义 (Hogan et al., 2021; Noy et al., 2019)，已有大量工作将其与语言模型结合 (Pan et al., 2024)：模型沿图路径推理 (Sun et al., 2024; Ma et al., 2025; Luo et al., 2024; Jiang et al., 2023)，检索子图作为上下文 (Baek et al., 2023; He et al., 2024; Mavromatis & Karypis, 2025)，或在文本上构建图索引以支持检索 (Edge et al., 2024; Gutiérrez et al., 2024; 2025; Guo et al., 2025; Liang et al., 2025; Li et al., 2025; Sarthi et al., 2024; Peng et al., 2025)。针对关系数据，基于本体的数据访问将本体映射到表，使查询能够用领域概念表达 (Poggi et al., 2008; Lenzerini, 2011; Calvanese et al., 2016; Das)，工业语义层则为所有使用方一次性定义度量、维度和连接 (dbt Labs, 2026; Google, 2026; Cube, 2026)；文本到SQL方法则针对原始模式生成查询 (Yu et al., 2018; Li et al., 2023; Lei et al., 2025; Pourreza & Rafiei, 2023; Gao et al., 2024; Talaei et al., 2024; Wang et al., 2025a)，在语义依赖业务知识的企业模式上仍表现较弱 (Chen et al., 2024; Liao et al., 2026; Gwimm & Eisenach, 2026)。在一项基准中，将企业问题表述在数据库的知识图谱上而非SQL模式上，使GPT-4的准确率从16%提升到54%；再依据本体检查查询，准确率提升到72%（两者均为作者自报；Sequeda et al., 2024; Allemang & Sequeda, 2024）；一项配对基准报告，三个前沿模型通过语义层查询时，错误与幻觉更少（作者自报；Rumiantsau & Fokeev, 2026）。本体也为检索提供依据 (Sharma et al., 2025)，受治理的分析接口将聚合逻辑移出模型 (Singh et al., 2026)，ChatDB则为智能体提供数据库作为符号记忆，通过SQL读写 (Hu et al., 2023)。与本文最接近的是EvoOntology：它让构建智能体为数据智能体构建本体层，并通过评估加以改进 (Chong et al., 2026)。这些工作都是按单个问题评估的读模型。我们加入写入路径和触发器，即以提交条件编码制度的动作，以及响应变化的自动化规则；由执行工作的智能体构建它们，并在长时间连续会话中评估，完整核算成本。

用语言模型构建知识图谱与本体。语言模型从文本中抽取实体、关系和模式 (Zhu et al., 2024; Zhang & Soh, 2024; Mo et al., 2025; Lairgi et al., 2024; Bai et al., 2026)，学习与构建本体 (Babaei Giglou et al., 2023; Lo et al., 2024; Zhang et al., 2025a; Oyewale & Soru, 2026)，并通过审核纳入事实 (Bykampadi et al., 2026)；基准依据参考图谱对图谱本身评分 (Mihindukulasooriya et al., 2023)，而本体工程通过模型能够回答的能力问题及建模陷阱目录评估，因为通常不存在单一的标准本体 (Grüninger & Fox, 1995; Poveda-Villalón et al., 2014)。本文的输入和成功标准均有所不同：智能体将结构化数据源与制度文本一同建模，模型除类别与关系外还包含操作和规则，并以其支持的工作及构建成本来评价。

上下文、记忆与长程智能体。长上下文的表现会随长度和信息位置而退化 (Liu et al., 2024a; Hsieh et al., 2024; Modarressi et al., 2025; Hong et al., 2025)，这推动了检索 (Lewis et al., 2020; Gao et al., 2023b; Asai et al., 2024) 及其与长上下文的结合 (Li et al., 2024; Jiang et al., 2024)。智能体记忆系统对智能体自身的经验进行分页、摘要或结构化组织 (Packer et al., 2023; Park et al., 2023; Shinn et al., 2023; Xu et al., 2025b; Chhikara et al., 2025; Wang et al., 2025d; Zhang et al., 2025c; Sumers et al., 2024; Zhang et al., 2024)，也包括将经验组织为时序或世界模型知识图谱 (Rasmussen et al., 2025; Anokhin et al., 2025)，并在长对话上进行评估 (Wu et al., 2025; Maharana et al., 2024; Hu et al., 2026)；对于精确回忆，由智能体控制的原始日志搜索可达到结构化记忆的水平 (Li et al., 2026)。企业与办公工作智能体基准测试数据库和应用中的工具使用 (Styles et al., 2024; Yao et al., 2025; Barres et al., 2025; Xu et al., 2025a; Wang et al., 2024d; Trivedi et al., 2024; Huang et al., 2025; 2026; Drouin et al., 2024; Boisvert et al., 2024; Wang et al., 2025c; Gruenbaum et al., 2026; Luo et al., 2025; Wang et al., 2026d; Liu et al., 2024b; Mialon et al., 2024)；在长时间模拟经营中，智能体会偏离目标，且表现方差很高，原因并不明显是上下文被填满 (Backlund & Petersson, 2025)。很少有工作在报告准确率的同时报告成本 (Kapoor et al., 2024; Wang et al., 2025b)。语义图而非经验记忆：它对领域的状态和规则建模，并与记录系统保持同步。OpsWEEK通过演化数据上的连续会话、持续指令和按状态判分，补充了现有基准。

工具、编译后的知识与自动化。智能体通过工具和 API (Yao et al., 2023; Schick et al., 2023; Patil et al., 2024; Qin et al., 2024; Yan et al., 2024; Anthropic, 2024) 或代码 (Wang et al., 2024b; Gao et al., 2023a; Wang et al., 2024e; Liang et al., 2023) 采取行动。多条研究路线将知识编译为可执行产物：语言模型编写规划领域或世界模型，随后由求解器或模拟器使用 (Liu et al., 2023; Guan et al., 2023; Tang et al., 2024; Wang et al., 2026b)；智能体积累可复用的工具和技能 (Wang et al., 2024a; Cai et al., 2024; Qian et al., 2023; Yuan et al., 2024; Wang et al., 2024c; Zheng et al., 2025; Hu et al., 2025)，也可将其纳入受治理的组织注册表 (Guo, 2026d)。制度也可编译为守卫，在运行时检查智能体的工具调用，来源可以是制度文档 (Zwerdling et al., 2025)、规则语言 (Wang et al., 2026a; Rebedea et al., 2023) 或经过验证的制度模型 (Chen et al., 2025; Xiang et al., 2025)。业务工件将操作描述为具有生命周期、以数据为中心的对象 (Nigam & Caswell, 2003; Bhattacharya et al., 2007)，其守卫-阶段-里程碑形式通过受守卫约束的转换与事件驱动生命周期 (Hull et al., 2011)，是带提交条件的动作及自动化规则最接近的形式化先例。事件-条件-动作规则与触发器是主动数据库中的经典机制 (McCarthy & Dayal, 1989; Widom & Ceri, 1995)，约束语言用于验证图数据 (Knublauch)。通过模型写回数据源属于视图更新问题 (Bancilhon & Spyrtos, 1981; Dayal & Bernstein, 1982)，基于本体的数据访问也研究了此问题 (De Giacomo et al., 2017)；实践中将命令的写入路径与根据事件重建的读模型分离 (Fowler, 2011; 2005)，我们的主机正是这样通过发件箱写入并重新同步。如今，语言模型生成工作流与流程自动化 (Ye et al., 2023; Zeng et al., 2023; Zhang et al., 2025b; Fan et al., 2025)，决定何时主动采取行动 (Lu et al., 2025; Liu et al., 2026)，并执行受治理的后台业务计划 (Moslemi et al., 2026)。在语义图中，操作、守卫与触发器并非独立代码：其类型由智能体查询的同一对象模型约束，引擎强制检查其条件，智能体则在与模式相同的分支-验证-发布周期中编写它们。

## I 声明

可复现性。 本文每个结果数值均由分析脚本根据运行记录（请求记录、计量日志及 ERP 变更日志）生成；每次运行记录其数据包、提示及系统提示的哈希值。提示和接口帮助在主研究前的一个 S 规模试运行会话上调试；附录 G 的判分修正发生在主研究之后。第 4-6 节及附录 B-G 描述了条件、提示、限制及判分。基准生成器、ERP 服务器、参考判定器、评分器、驱动程序和分析代码可向作者获取；本体内核属于商业平台，未发布。

伦理。 本研究不涉及人类参与者。全部基准数据均为合成，其中的公司为虚构公司。第 1 节的生产观察是作者自身部署中的汇总计数，不包含客户数据。智能体在设有内存、进程及网络限制的沙箱中运行，与生产智能体共享运行时池；监控程序在池的可用内存下降时暂停研究。

语言模型的使用。 DeepSeek V4.1 Flash 是实验研究对象。编码与写作助手帮助实现实验代码、筛选文献及翻译中文版本；作者依据运行记录和来源核查了每项论断、数值及参考文献。

利益冲突与资助。 作者受雇于 AIDC，该公司开发图条件使用的本体内核，并资助了本研究。我们引用了本团队此前的三篇工作论文 (Guo, 2026a;c;b)，以及一项关于受治理自主改进的研究 (Guo, 2026d)。