arXiv:2603.09152v1 [cs.AI] 10 Mar 2026

亮点

DataFactory：用于高级表格问答的协作式多智能体框架

Tong Wang、Chi Jin、Yongkang Chen、Huan Deng、Xiaohui Kuang、Gang Zhao

- 多智能体框架通过专业化团队协调，革新表格问答
- 自动化的数据到知识图谱转换借助映射函数实现多跳推理
- 数据库团队和知识图谱团队的准确率分别提升了20.2%和23.9%
- 自然语言咨询增强协作；上下文工程减少幻觉
- 平台提供直观的界面，用于数据探索和可视化

DataFactory：用于高级表格问答的协作式多智能体框架 ⋆⋆

Tong Wang[a,1], Chi Jin[b,1], Yongkang Chen[a], Huan Deng[a], Xiaohui Kuang[a] 和 GangZhao[a,*]

[a]军事科学院系统工程研究所，中国北京，*100101*[b]中国人民大学信息资源管理学院，中国北京，*100872*

文献信息

关键词：表格问答 多智能体系统 大语言模型 知识图谱 数据工厂 ReAct范式

摘 要

表格问答（TableQA）支持通过自然语言与结构化表格数据进行交互。然而，现有的大语言模型（LLM）方法面临几个关键局限：上下文长度限制约束了数据处理能力，幻觉问题损害了答案的可靠性，而单智能体架构难以应对涉及语义关系和多跳逻辑的复杂推理场景。本文提出 DataFactory，这是一种通过专业化团队协作和自动化知识转换来解决上述局限的多智能体框架。该框架由采用 ReAct范式进行推理编排的数据负责人，以及专门的数据库团队和知识图谱团队组成，能够将复杂查询系统地分解为结构化推理和关系推理任务。我们通过映射函数 $\mathcal{T} : \mathcal{D} \times \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{G}$ 对自动化的数据到知识图谱转换进行形式化，并实现基于自然语言的协商；与固定工作流的多智能体系统不同，该机制支持灵活的智能体间协商和自适应规划，从而提升协作的稳健性。此外，我们还采用上下文工程策略，整合历史模式和领域知识，以减少幻觉并提高查询准确性。在 TabFact、WikiTableQuestions 和 FeTaQA 数据集上，使用来自五家提供商的八个大语言模型进行的实验均显示出稳定的性能提升。与基线方法相比，我们的方法将准确率提高了 20.2%（TabFact）和 23.9%（WikiTQ），且效果显著（Cohen's d>1）。团队协作的表现也优于单团队变体（TabFact 提高 +5.5%，WikiTQ 提高 +14.4%，FeTaQA ROUGE-2 提高 +17.1%）。该框架为多智能体协作提供了设计指南，并通过整合结构化查询与基于图的知识表示，为企业数据分析提供了实用平台。

1. 引言

表格问答（TableQA）使用户能够针对结构化表格数据提出自然语言问题，从而提取相关信息和见解（Jin等，2022 年）。与传统的基于文本的问答（QA）不同，表格问答需要理解表格结构，包括行、列、单元格内容和语义关系，才能准确检索答案。借助表格资源，这项任务能够实现对复杂结构化信息的自然语言查询，并在各种场景中具有实际应用价值。

表格问答的早期研究主要采用基于规则和基于检索的方法（KatyayanandJoshi，2022 年；Wei等，2006），随后发展出深度学习方法，例如 TAPAS、TAPEX 和 TaBERT，这些方法联合编码自然语言和表格模式（Herzig等，2020；Yin等，2020）。尽管这些方法取得了进展，但在处理多跳查询和语义关系推断时，仍面临可解释性和复杂推理方面的挑战。

近期进展将大语言模型（大语言模型）引入表格问答，采用直接提示和代码生成范式（Lu 等人，2025；Fang 等人，2024年）。然而，现有方法面临诸多关键挑战，包括上下文长度限制、幻觉问题，以及在涉及语义关系和多步逻辑的复杂推理场景中难以应对（Nan 等人，2022 年；Sui 等人，2024年）。

⋆本研究得到中国国防科技重点实验室基金（资助编号：6142006220506）的支持。⋆⋆本稿为发表于《信息处理与管理》的文章的录用版本。© 2026 爱思唯尔有限公司。保留所有权利。

*通讯作者 tongwss@foxmail.com (T. Wang); jinchi997@163.com (C. Jin); yk.chen365@outlook.com (Y. Chen); DHu[illegible]56@foxmail.com (H. Deng); xhkuang@bupt.edu.cn (X. Kuang); bisezhaog@163.com (G. Zhao) ORCID: 0000-0001-6981-916X (T. Wang); 0009-0009-4010-2718 (C. Jin); 0009-0006-8668-777X (G. Zhao) [1]这些作者对本研究做出了同等贡献。

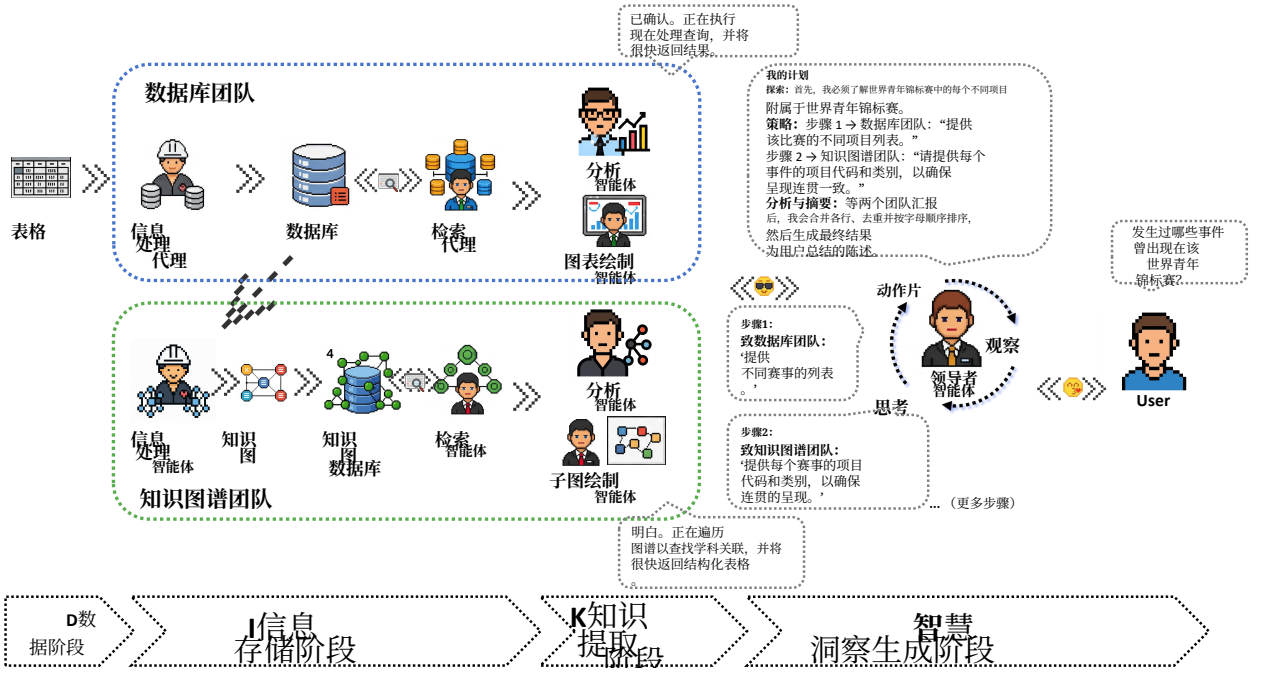


图 1： 基于大语言模型的多智能体 DataFactory 框架的表格问答概览。该系统由数据库团队、知识图谱团队和数据负责人组成，数据负责人负责协调各团队协作，从初始数据阶段开始，依次完成三个处理阶段：信息存储、知识提取和洞察生成。

为解决这些问题，我们提出一种用于表格问答的基于大语言模型的多智能体 DataFactory 框架，该框架：（1）借助大语言模型辅助的模式理解，实现表格数据摄取自动化；（2）通过映射 $\mathcal{T} : \mathcal{D} \times \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{G}$ 将数据到知识图谱的转换形式化，以捕获语义关系；（3）通过上下文工程和少样本提示减少幻觉，其中整合了历史问答、数据定义语言（DDL）/图模式及领域知识；（4）在完全自主的流水线中联合利用结构化检索（结构化查询语言（SQL））和关系检索（Cypher），实现多步推理。

所提出的 DataFactory 框架由一名数据负责人和两个专家团队（数据库团队和知识图谱团队）组成，采用三阶段工作流。在信息存储阶段，信息处理智能体会分析表头、字段名和类型，将数据导入数据库，并利用我们的规则和算法从表格中自动构建知识图谱。在知识提取阶段，检索智能体利用历史问答记录、DDL/图模式和领域知识，设计上下文工程提示并生成 SQL/Cypher 查询。随后，分析智能体会解读其输出，并在适当情况下将其可视化为统计图表或子图。在洞察生成阶段，数据负责人运用 ReAct 范式，将用户问题分解为多个子任务（数据探索、策略制定、答案综合），分派给各团队，并整合结果以形成连贯的响应。

总而言之，我们的贡献包括：（1）专业团队协调机制，通过建立专门的数据库团队和知识图谱团队，将表格问答任务系统地分解为互补的推理模式，并通过基于自然语言的协商和自适应策略调整突破单智能体局限；（2）通过形式化的数据到知识图谱转换实现自动化知识集成 $\mathcal{T} : \mathcal{D} \times \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{G}$，从而支持一致的实体消歧、语义关系发现，以及面向实际部署场景的可扩展系统架构；（3）由数据负责人采用 ReAct 范式进行动态推理编排，以协调结构化检索与关系推理。在此框架下，我们通过与代表性基线方法进行比较，并在跨提供方、跨规模模型设置下开展协作模式分析和消融研究，进行系统性实验，综合评估性能与可扩展性。该框架还配有一个交互式平台[2]，提供全面的数据探索界面，

[2]平台演示网站：https://wisdomindata.netlify.app/。该网站展示交互式演示视频，并将在集成完成后托管开源代码。

知识图谱可视化和交互式多智能体协作，使用户能够分别与数据库团队和知识图谱团队进行交互，或通过数据负责人编排联合工作流。

2. 相关工作

2.1. 基于大语言模型的表格问答方法

目前，大语言模型主要通过两种范式应用于表格问答：直接提示方法和代码生成方法（Lu 等，2025；Fang 等，2024）。

直接提示方法。直接提示是指将表格内容（通常经过适当格式化）与问题一同作为上下文提供给大语言模型，让模型直接生成答案。然而，这种方法面临显著挑战：（1）上下文长度限制会约束数据处理能力（Wang 等，2025a；Lee 等，2025）；（2）存在幻觉问题，即模型生成表格数据不支持的内容（Farquhar 等，2024；Yang 等，2025c；Sairaj 和 Balasundaram，2025a）；（3）难以精确定位数据，且端到端推理过程的可解释性较低，因此难以明确说明答案来源或推导过程（Nguyen 等，2025；Zhao 等，2023；Zhang 等，2024c）。Sairaj 和 Balasundaram（2025b）及 Sairaj 和 Balasundaram（2025a）的近期研究表明，大语言模型往往捕捉的是词汇共现模式，而非真正的实体关系，导致生成内容出现事实性幻觉和主题冗余。一些方法尝试通过微调提高准确率，但在解决这些核心挑战方面成效有限，同时还增加了实际应用的复杂性（Liu 等，2022；Zhang 等，2024a；Yang 等，2025b）。

代码生成方法。第二种范式涉及大语言模型生成可执行的程序代码，将问答任务转换为通过外部工具操纵表格的可执行操作，从而提供透明性和可验证性。该范式包括：(1) Python 代码生成，利用 pandas 等库进行筛选和计算（Evangelatos 等，2025；Zhang 等，2024b），但这种方法依赖运行时环境，并且在处理分层表格结构或稀疏数据分布时性能会下降（Cao 等，2023）；(2) 电子表格公式生成，即大语言模型生成公式，这要求准确的单元格引用并遵循公式语法——执行通常需要与 Excel 交互，从而增加了实现的复杂性（Wang 等，2025b）；以及 (3) SQL 查询生成，将自然语言转换为 SQL 查询，但需要进行数据库模式设计，并且在多步推理任务中存在局限性（Nguyen 等，2025；Chen 等，2021；Qin 等，2022；Pal 等，2024；Guo 等，2025）。

总的来说，这两种范式都强调结构化字段提取，但在处理涉及语义关系、跨行综合或多行数值比较的复杂推理问题时仍有局限（Nan 等人，2022 年；Sui 等人，2024年）。这些局限促使研究者探索基于智能体的框架，通过引入显式推理和行动机制来应对。

2.2. 基于智能体的表格问答方法

基于智能体的方法将复杂推理分解为推理与行动的迭代循环，从而与单次处理的大语言模型方法相比，提高可解释性并减少错误。现有研究探索了多个技术方向：

ReAct 风格分解。受REACT 范式（Yao 等人，2023年）启发，这些框架交替执行推理步骤与工具调用，以系统地浏览表格（Zhang 等人，2024b）。典型工作流包括识别相关行或列、执行 SQL/Python 查询、反思中间输出，并不断完善结果，直至确认答案。管理多跳推理的复杂性并确保反思机制可靠，仍是关键挑战。

单智能体优化。 已有多种单智能体框架被提出，通过迭代优化和自我纠正机制来提升推理质量。这些方法通常采用基于角色的推理（Zhao等，2025）或多步验证来提高答案准确率（Yu等，2025）。尽管这些方法能有效减少某些类型的错误，但在处理需要多元专业知识和并行处理能力的复杂查询时仍存在局限。

知识增强。一些方法通过整合外部知识源或基于图的表示来增强推理能力（Zhao 等人，2025；Li 等人，2025；Meier 等人，2025）。然而，这些方法通常侧重于部分表格内容或浅层的表间链接，对语义关系的覆盖有限。它们对外部知识的整合往往仍停留在表面，因此难以进行深入的多跳推理。

工具编排与规划。基于规划的系统会有选择地调用异构工具，例如使用 SQL 进行检索、使用大语言模型进行推理（Khoja 等人，2025；Pal 等人，2024；Ojuri 等人，2025）。这提高了透明度和效率，但在维持全局一致性以及从执行失败中恢复方面仍面临挑战。近期的框架（如 **StructGPT** ）为表格、知识图谱和数据库设计了接口（Jiang 等人，2023年）。然而，这些接口仍各自依附于独立的数据源，而未形成用于统一表格问答的集成工作流。

多智能体协作。近期研究采用具有明确角色专门化的多智能体流水线。例如，**AutoPrep** 将数据准备分解为规划器、程序员和执行器等角色（Fan 等人，2025）。**AutoTQA** 进一步拓展了这一思路，通过协调多个大语言模型智能体实现端到端表格问答（Zhu 等人，2024年）。类似地，**MACT** 表明，将规划和编码智能体与外部工具相结合，无需微调也能实现有竞争力的准确率（Zhou 等人，2025）。

然而，Pan 等人（2025）近期的实证分析显示，多智能体系统往往无法胜过单智能体基线，其失败主要归因于协调问题、规范违规和验证机制不足，而非个别大语言模型的局限。他们提出的多智能体系统失败分类法（MAST）指出，在规范设计、智能体间失配和任务验证方面存在 14 种不同的失败模式。这表明，当前方法需要的是根本性的架构改进，而非战术性的提示工程修补。

尽管存在这些挑战，现有系统凸显了分工的潜力，但其重点仍主要在于表格操作和数据库查询，协调也主要依赖预先设定的工作流执行，而非动态协商与推理。我们的 **DataFactory** 框架通过引入专门的知识图谱团队，推动了这一研究方向，使数据能够转换为知识图谱，并支持 SQL 查询与图查询之间的集成推理。更重要的是，我们的方法支持数据负责人、数据库团队和知识图谱团队这三个专业智能体之间基于自然语言的协商与审议，而非采用僵化的、基于工作流的任务分配，从而实现更加灵活且自适应的问题求解策略。

从适用场景来看，固定程序式多智能体框架更适合步骤明确、可预测且预算严格的流程，能够实现确定性的执行并保持较高的稳定性。相比之下，正如我们的框架所采用的，基于自然语言的协作更适用于复杂或探索性查询；这类查询需要自适应规划、跨 SQL 查询与图查询的多跳推理，以及基于中间证据动态调整策略。这两种范式互为补充；我们的研究针对的是后一种场景。

总之，尽管基于智能体的表格问答方法展现出令人期待的能力，但仍存在若干关键挑战，促使我们设计多智能体数据工厂：

（1）**缺乏专业化团队协调**：现有方法主要依赖角色专门化程度有限的单智能体系统，这限制了结构化数据处理与关系知识表示之间互补优势的发挥。

（2）**知识集成不足**：当前知识增强方法侧重于部分表格内容或表层的表间关系，缺乏对数据关系以及实体间深层语义联系的全面理解。

（3）**推理编排不充分**：当前框架难以动态协调不同的数据访问模态，错失了通过结合结构化检索与关系推断来开展复杂多步推理的机会。

（4）**自然语言协商机制有限**：现有多智能体系统主要采用僵化的基于工作流的协调方式，缺乏灵活的自然语言协商与审议能力，因而无法在复杂推理过程中实现自适应策略调整和跨团队知识共享。

（5）**自动化与可扩展性有限**：很少有框架支持系统化的数据摄取或知识图谱构建，限制了其在现实世界中的部署。

3. 研究目标

基于上述分析，本文旨在提出一套全面的解决方案，在推进表格问答多智能体推理的同时，针对这些局限性。具体而言，我们的研究通过以下目标，聚焦于专业化、协调和可扩展性方面的根本问题：

(1)建立专业团队协调机制：我们提出组建专门的数据库团队和知识图谱团队，发挥结构化数据处理与关系知识表示的互补优势，突破单智能体局限，实现系统化的任务分解。这种方法旨在

通过明确角色边界并制定自然语言协商协议，缓解 MAST 所指出的协调失效问题。

(2)**实现自动化知识集成**：我们的目标是通过自动化数据到知识图谱转换算法实现稳健的跨源对齐，确保实体消歧和语义集成的一致性，从而实现全面的数据理解。

(3) **设计复杂的推理编排**：我们引入一位数据负责人，采用 ReAct范式，在结构化检索与关系推理之间进行动态协调，从而实现复杂的多跳推理。

(4) **实现基于自然语言的协商机制**：我们开发了一种灵活的智能体间通信机制，使专业团队能够通过自然语言进行协商和审议，从而在复杂推理过程中促进自适应策略调整和知识共享，而非依赖僵化的工作流执行。

(5) **提升系统可扩展性**：我们的目标是开发完全自主的数据摄取和知识图谱构建流水线，以应对复杂的表格问答场景，包括多跳推理、语义关系分析，以及实际部署中的知识集成。

4. 方法论

本节全面阐述我们提出的多智能体数据工厂的内部机制和核心算法。我们首先在第4.1节概述系统的协作架构，提供各个核心组件的总体蓝图。随后各节将逐一探讨各组件的具体实现：第4.2 节详述数据库团队如何处理结构化数据并检索目标信息及相关洞见；第4.3 节介绍知识图谱团队如何进行关系知识表示与推理；第4.4 节则说明数据负责人如何分解问题解决流程，并协调两个团队以生成全面的洞见。

4.1. 概述

DataFactory采用三方协作架构，通过专业专长与智能协调来分解复杂的表格问答任务。如图1所示，该框架通过三个协调运作的组件实现我们的研究目标：负责统筹基于自然语言的协商机制的数据负责人、专注于结构化数据处理的数据库团队，以及专注于关系知识表示的知识图谱团队。

数据负责人采用 ReAct 范式，将用户查询分解为结构化推理和关系推理子任务，从而实现复杂的推理编排。数据负责人不遵循预先设定的工作流程，而是与专家团队进行动态协商，根据查询复杂度和中间结果自适应地调整策略。这种基于自然语言的协调机制，代表了相较于现有多智能体系统中僵化任务分配方式的一项重要进步。

数据库团队通过上下文增强 SQL 生成，为结构化数据操作建立专门的协调机制，并整合历史问答模式、模式信息和领域知识。该团队的核心创新在于将自动化数据摄取与检索增强生成技术相结合，在保持数值计算和聚合查询准确率的同时，减少幻觉。

知识图谱团队通过我们正式化的数据到知识图谱转换 $\mathcal{T} : \mathcal{D} \times \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{G}$，实现自动化知识集成：系统地将表格数据 $\mathcal{D}$、模式定义 $\mathcal{S}$ 和关系模式 $\mathcal{R}$ 转换为知识图谱 $\mathcal{G}$。这种自动化转换解决了可扩展性限制，使系统能够系统地发现语义关系并执行多跳推理，从而超越传统的结构化查询能力。

该框架通过三个相互协调的阶段，将原始数据转化为可执行的洞察。在信息存储阶段，自动化数据摄取借助大语言模型辅助的模式理解和关系发现，构建结构化数据库与语义知识图谱。知识提取阶段采用上下文增强提示策略，整合历史模式、模式信息和领域知识，生成准确的 SQL 和 Cypher 查询，同时降低幻觉风险。洞察生成阶段通过基于 ReAct 的分解协调多智能体协作，由数据负责人动态分配推理子任务，并从结构和关系两个视角综合结果。

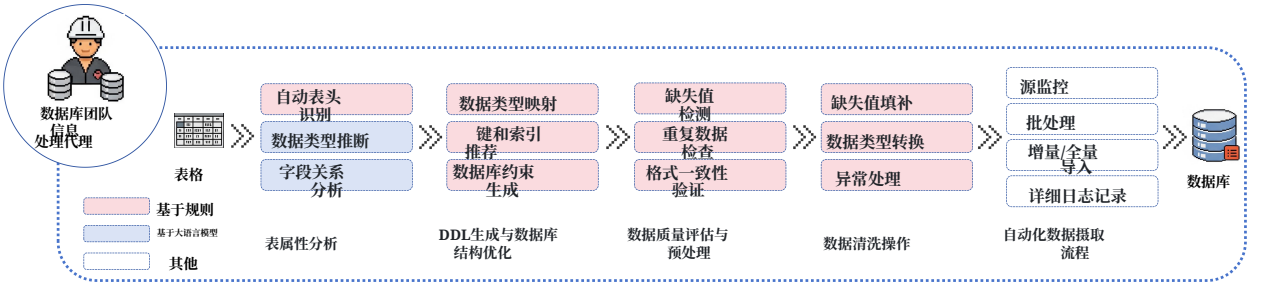


图2： 数据库信息处理代理的工作流程。该自动化流水线将用于模式构建的基于规则的操作与用于语义理解的基于大语言模型的分析相结合，涵盖表格分析、DDL生成、数据摄取和质量评估。

这种架构设计解决了在统一框架中结合互补推理模式这一根本挑战。通过将表格问答解耦为结构化查询和关系推理能力，该框架能够进行复杂的多跳推理，发挥两种范式各自的优势，同时通过明确的协调机制保持可解释性。

4.2. 数据库团队

数据库团队建立了专门的协调机制来处理结构化数据操作，凭借在数值计算、聚合和精确数据筛选方面的专业能力，实现了第一个研究目标。该团队的架构创新在于将大语言模型辅助的语义理解与传统数据库操作相结合，在实现自动化数据摄取的同时，通过上下文增强提示策略确保查询准确性。

该团队由四个专业化智能体组成，共同实施我们的方法，在降低幻觉风险的同时保持计算效率。信息处理代理通过大语言模型与规则混合集成实现自动化数据摄取；信息检索代理利用检索增强提示实施上下文增强的文本到SQL生成；信息分析代理对结构化结果进行语义解读；可视化智能体则支持分析洞见呈现。这种专业化直接解决了单智能体方法在处理多样化结构化数据处理要求方面的局限。

4.2.1. 数据库信息处理代理

数据库信息处理代理采用混合方法，通过结合基于大语言模型的语义理解与基于规则的优化，实现结构化数据的自动化知识集成。该设计通过在无需手动模式工程的情况下实现系统化数据摄取，同时保持数据完整性和查询性能，解决了研究目标中指出的可扩展性限制。

该智能体的核心创新在于智能模式理解：利用大语言模型对异构表格格式进行语义解析，同时采用统计分析和基于规则的验证来确保稳健性。与仅依赖启发式规则的传统数据处理流水线不同，我们的方法通过语义上下文分析，能够自适应地处理不一致的表头格式和含糊的数据类型。大语言模型组件会分析表格结构和内容语义，以识别实体关系和数据依赖项；基于规则的组件则确保句法正确性和约束满足。

自动化DDL生成流程是将语义理解与数据库模式设计相结合的一项重要进展。智能体会分析观测到的数据特征，推荐适当的约束、索引和关系，从而减少人工模式工程工作量，同时保持查询优化特性。

质量保证机制将基于大语言模型的判断与预设验证规则相结合，以处理数据不一致和缺失值模式。该智能体根据数据特征和领域需求调整清洗策略，提供静态的基于规则方法无法实现的灵活性。这种自适应质量控制有助于确保数据的一致性和完整性（见图 2）。

数据库团队检索智能体提示模板：

智能体档案提示词：	你是一位 SQL 分析师，专注于系统化的数据探索与发现。 请帮助生成 SQL 查询来回答这个问题。
正在工作原则提示词：	**===核心使命===** - 理解用户真正想要什么 - 在做出假设之前，先了解数据的实际情况 - 逐步从宽泛查询深入到具体查询 - 当方法失效时调整策略 **===探索策略 ===** 始终：发现 → 采样 → 定位目标 → 优化 切勿：未经验证就假定值存在 **===响应指南===** 1. 充分的上下文：生成有效的 SQL，无需解释 2. 部分上下文：为数据发现创建中间 sql 3. 上下文不足：说明为何无法生成查询 4. 相关性：使用最相关的表格和列 5. 一致性：对先前提出的问题重复给出完全相同的答案 6. 合规性：确保语法符合 PostgreSQL 规范且可执行
表结构与 DDL:	**===表信息===** 创建表 customers (id 整数 主键, -- 唯一的客户标识符，自增整数 name 可变长度字符串(100), -- 客户的全名，包括名和姓 email 可变长度字符串(100), -- 客户的联系电子邮箱，用于沟通和账户识别 sales_amount 定点数(10,2), -- 客户的累计购买金额，最多支持 8 位整数和 2 位小数 地区 可变长度字符串(50), -- 客户的地理地区或销售区域 创建日期 日期 -- 客户账户创建的日期，格式为 年-月-日);
附加上下文:	**===附加上下文===** [业务规则和领域知识在此处] **===示例===**
历史少样本示例:	用户：按销售额排名，我们排名前 5 的客户是哪些? 助手：SELECT name, sales_amount FROM customers WHERE sales_amount IS NOT NULL ORDER BY sales_amount DESC LIMIT 5; ...
问题:	针对[表名], [当前问题]

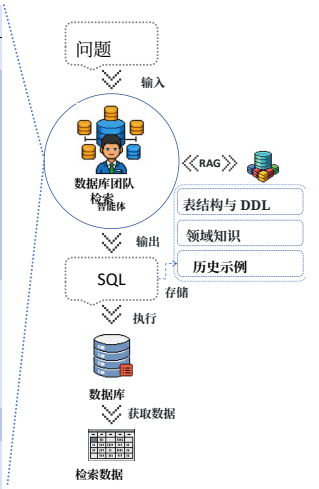


图 3： 数据库信息检索代理的架构。该代理整合表结构和 DDL 信息、领域知识以及检索到的历史问题-SQL对，以构建用于 SQL生成和数据库查询的提示词。

4.2.2. 数据库信息检索代理

数据库信息检索代理通过多源知识整合，实现上下文增强的文本到SQL生成，以应对幻觉风险。该智能体将历史模式、模式信息和领域知识系统地融入查询生成流程，体现了复杂推理编排方面的一项重要进展。

该智能体的核心创新在于自适应上下文构建，它通过检索增强提示策略扩展了传统的文本到SQL方法。该智能体不依赖静态示例或仅基于模式的指导，而是从三个互补来源动态汇集上下文信息。历史问答检索利用语义相似度，从经过验证的问题-SQL对中识别相关查询模式，使系统能够借鉴成功的推理先例，同时避免重复以往的错误。为了持续构建这一历史知识库，系统会在每次交互后自动保存每个用户查询及其对应的生成SQL和执行结果，并将其作为向量化表示存储在专用向量数据库中，以便高效进行语义检索。模式感知集成通过全面纳入DDL，确保生成的查询与实际数据库结构一致；领域知识注入则支持对业务术语和计算约定进行精细解读。

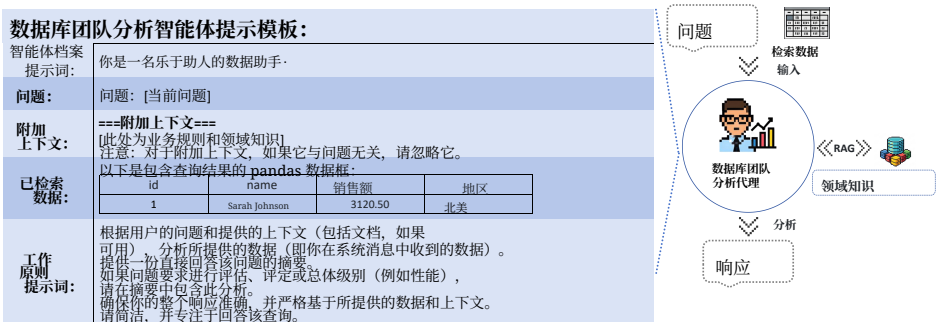
检索增强生成（RAG）机制是减少SQL生成中幻觉的重要方法学贡献。与随机选择示例或使用预设模板的传统方法不同，我们的智能体采用语义嵌入和结构兼容性评分来识别与上下文相关的先例。这种自适应选择流程依据语义相关性和句法模式兼容性筛选示例，确保少样本提示能够为当前查询生成任务提供有意义的指导和准确的结构模板。

多源信息融合使智能体能够生成同时满足多项正确性标准的 SQL 查询。该综合流程确保与用户意图的语义一致性、针对目标数据库模式的句法正确性，以及与特定领域业务要求的逻辑一致性（见图 3）。

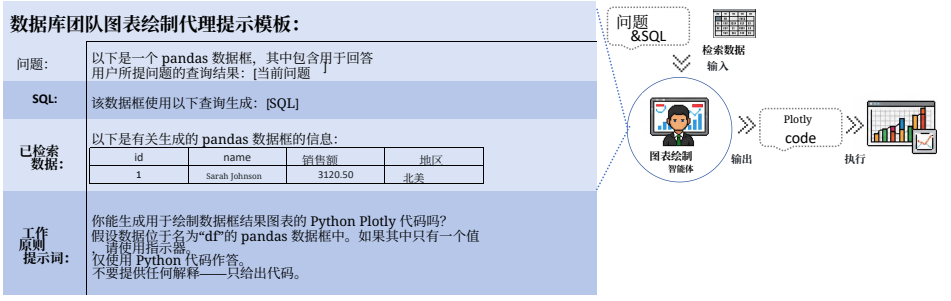
4.2.3. 数据库信息分析代理和可视化智能体

数据库信息分析代理负责解读 SQL 查询结果，并为终端用户生成自然语言解释。该智能体解决了传统数据库系统的局限性：传统系统仅返回原始表格数据，而不提供上下文解释；数据库信息分析代理使用户能够在更广泛的分析上下文中理解查询结果的语义。

智能体通过系统化的数据解读处理结构化查询结果，识别关键指标、维度和统计模式。对于涉及聚合、分组或排序操作的分析型查询，



（a）带有提示模板的数据库信息分析代理



(b) 带有提示模板的数据库可视化智能体

图4： 数据库信息分析与可视化智能体。(a) 分析代理将检索到的表格转换为结合领域上下文的自然语言摘要。(b) 可视化智能体生成绘图代码，以生成用于数据探索的交互式图表。

智能体会分析这些计算的业务意义，并将其与原始用户查询的上下文相结合。这一解读流程不止于简单的数据呈现，还能提供有意义的分析洞见。

自然语言响应生成会将查询结果与用户意图相结合，形成连贯的解释，总结发现并提供相关背景信息。智能体不会只返回孤立的数值，而是会构建包含支持性证据和比较信息（如适用）的响应。对于数值结果，智能体会执行基本的统计计算，并识别值得关注的趋势或异常，以增进用户的理解。

智能体通过整合多行结果中的信息，满足跨行分析要求，从而得出全面的结论。这项能力支持复杂的商业智能查询，这类查询需要综合不同数据维度中的模式，使用户能够理解数据中显现的关系和趋势。

当可视化有助于理解数据时，可视化智能体会通过生成适当的图表来补充分析流程。智能体会评估查询结果和用户要求，以确定最佳的可视化方式，然后生成绘图代码，通过 Web界面进行交互式展示（见图 4）。

4.3. 知识图谱团队

知识图谱团队通过关系数据表示实现自动化知识集成，将扁平表格数据转化为支持多跳推理和关系发现的语义网络，从而应对第二项研究目标。当用户问题需要理解数据元素之间的实体关系和间接关联时，该团队专门从事基于图的查询处理。

团队由四个专业智能体组成，分别负责基于图的推理中的不同环节：信息处理智能体从表格数据构建知识图谱，信息检索智能体生成用于图遍历的 Cypher 查询，信息分析智能体解读图查询结果，可视化智能体则将知识图谱呈现为可视化形式。这种分工能够系统地处理以关系为中心的查询，覆盖传统结构化数据聚合无法处理的内容。

4.3.1. 知识图谱信息处理智能体

知识图谱信息处理智能体通过将表格数据系统地转换为语义知识图谱，实现自动化知识集成。该智能体旨在解决现有表格问答方法的一项根本局限：无法捕捉并利用实体之间的语义关系来进行复杂的多跳推理。

我们的关键创新在于通过映射函数，将自动化的数据到知识图谱转换过程形式化：

$$\mathcal{T} : \mathcal{D} \times \mathcal{S} \times \mathcal{R} \to \mathcal{G} \tag{1}$$

其中，$\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ 表示输入的表格数据集，$\mathcal{S}$ 表示实体模式，用于指定如何从表格行中识别和提取实体；$\mathcal{R}$ 表示关系规则，用于确定实体如何在语义上相连；$\mathcal{G} = (V, E, A)$ 表示生成的知识图谱，其中包含节点 V、边 E和属性 A。不同于依赖手动模式工程或启发式规则的现有方法，我们的转换方法结合大语言模型辅助模式识别与算法验证，实现了系统化的语义关系发现。

这一流程遵循“识别—构建—连接—优化”范式，兼顾效率与准确率。从概念上看，它包含两个阶段：实体构建和关系发现。在实际应用中，实施过程分为三个工作流阶段：策略生成、配置验证和图构建。

(1) 实体构建理论框架

实体标识符生成机制。 为确保实体的唯一性和可追溯性，我们定义了标识符（ID）生成函数：

$$\text{ID}(row, type) = \begin{cases} type \oplus \text{Extract}(row), & \text{basic mode} \\ namespace \oplus type \oplus \text{Extract}(row), & \text{with namespace} \end{cases} \tag{2}$$

其中，$\text{Extract}(row)$ 从数据行中提取标识符值，$\oplus$ 表示使用适当的分隔符进行字符串拼接。对于单个标识符列，$\text{Extract}(row)$直接返回该列的值。对于多个标识符列，则用冒号将它们拼接起来，形成复合标识符。

基于拆分的实体构建策略。对于包含多个值的复合单元格数据，我们定义：

$$\text{Entities}(row, column) = \begin{cases} \{\text{Split}(row[column], \delta)\}, & \text{if delimiters detected} \\ \{row[column]\}, & \text{otherwise} \end{cases} \tag{3}$$

其中，$\text{Split}(value, \delta)$ 使用分隔符集合 $\delta = \{“,”、“;”、“|” \ldots\}$ 拆分单元格值，将其划分为独立的实体实例。例如，一个包含“人工智能,机器学习,数据科学”的单元格会生成三个独立的技术实体。

实体属性构建。对于每个实体，我们从多个来源汇总属性：

$$\text{Attributes}(entity) = A_{core} \cup A_{custom} \cup A_{meta} \tag{4}$$

其中：

- A_{core} 包含用于实体可追溯性的主要标识符属性
- A_{custom} 包含映射到该实体的用户指定属性列
- A_{meta} 存储系统生成的元数据（时间戳、来源信息等）

实体合并策略。 检测到重复实体时，我们会应用智能合并函数：

$$\text{Merge}(e_1, e_2) = (id_1, \text{Combine}(A_1, A_2), R_1 \cup R_2) \tag{5}$$

其中，$\text{Combine}(A_1, A_2)$ 实现基于优先级的属性合并策略：非空值会覆盖空值；对于非空值之间的冲突，则保留首次出现的值并记录冲突日志。

(2) 关系发现与建立理论框架

行内关系匹配机制。 行内关系匹配是最直观的关系建立模式，能够在单个数据行内不同实体之间建立语义联系。该机制的工作原理基于“共现即相关”的假设：出现在同一数据行中的不同类型实体之间存在某种语义关联。

对于数据行 d_i，我们将行内关系生成定义为：

$$R_{\text{same}}(d_i) = \{(e_s, e_t, r) \mid e_s, e_t \in \text{Entities}(d_i), \text{type}(e_s) \neq \text{type}(e_t), r \in \mathcal{R}_{\text{intra}}\} \tag{6}$$

其中 $\text{Entities}(d_i)$ 返回从行 d_i 中提取的所有实体， $\mathcal{R}_{\text{intra}}$ 定义不同实体类型之间允许的行内关系类型。

跨行关系匹配与分组策略。 跨行关系匹配能够发现超出单行数据范围的复杂语义联系，包括层级关系、依赖关系和时间关系。系统采用基于分组的跨行匹配策略，定义如下：

对于跨行关系发现，我们首先将数据集划分为逻辑组：

$$\text{Group}_k = \{d_i \mid d_i[\text{col}] = \text{value}_k, \forall \text{col} \in C_{\text{group}}\} \tag{7}$$

其中 C_{group} 表示分组列集合，值$_k$ 是组 k 的共享值。该分组策略基于以下原则：共享相似上下文属性的实体更可能具有有意义的关系。

多模式规则评估引擎。跨行关系的建立依赖于复杂的规则评估机制，系统支持两种规则类型：对于基于语义相似度的关系，我们定义：

$$\text{Similarity}(e_1, e_2) = \frac{\text{embed}(e_1) \cdot \text{embed}(e_2)}{||\text{embed}(e_1)|| \cdot ||\text{embed}(e_2)||} \tag{8}$$

A 当相似度$(e_1, e_2) \geq \theta$时建立关系，其中 θ 是预定义的阈值 d.

对于基于属性的比较，我们在对应的实体属性之间使用标准比较运算符（=、 >、 <、 ≥、 ≤）。

复合逻辑表达式处理。在实际应用中，关系建立通常需要满足复杂的逻辑条件组合。系统通过逻辑运算符（AND、OR）和分组机制，支持构建复杂的条件表达式。其运行机制采用递归评估策略：对于包含多个条件的规则集，系统根据逻辑运算符的优先级和分组结构进行递归评估，支持任意深度的逻辑嵌套。

转换流程需要结构化输入规范，包括表格数据 $\mathcal{D} \in \mathbb{R}^{n \times m}$、实体模式 $\mathcal{S} = \{(\tau_i, C^{(i)}$ id $C^{(i)}$ attr·split_配置$^{(i)})\}$ （用于定义实体抽取规则）以及关系规则 $\mathcal{R} = \{(r_j, \tau_{\text{src}}^{(j)}, \tau^{(j)}$ tgt·match_mode$^{(j)}$, 规则$^{(j)})\}$ （用于指定语义连接模式）。生成的知识图谱$\mathcal{G} = (V, E, A)$ 提供结构化语义表示，支持复杂查询操作和多跳推理。

实现遵循三阶段工作流（图5），在自动化与质量保证之间取得平衡。策略生成阶段利用基于大语言模型的分析，结合表格模式和样本数据，确定最优的实体模式 $\mathcal{S}$ 和关系规则 $\mathcal{R}$。这避免了手动配置，同时保持语义准确性。配置验证可确保算法一致性和参数有效性。图构建并行应用实体识别和关系发现算法，并通过优化的批处理操作将结果导出至 Neo4j（Neo4j，2012）。

这种设计实现了自动化知识集成，突破了手动模式工程的局限，同时保留了复杂表格问答场景所需的丰富语义。该方法能够系统地发现实体关系，而传统的结构化查询方法无法捕捉这些关系，从而支持高级问答任务所必需的多跳推理能力。

策略生成阶段利用表格模式和样本数据进行结构感知分析，并通过大语言模型分析实现语义理解。与启发式方法不同，该阶段会根据实际数据特征生成精确的实体提取和关系发现配置。配置验证

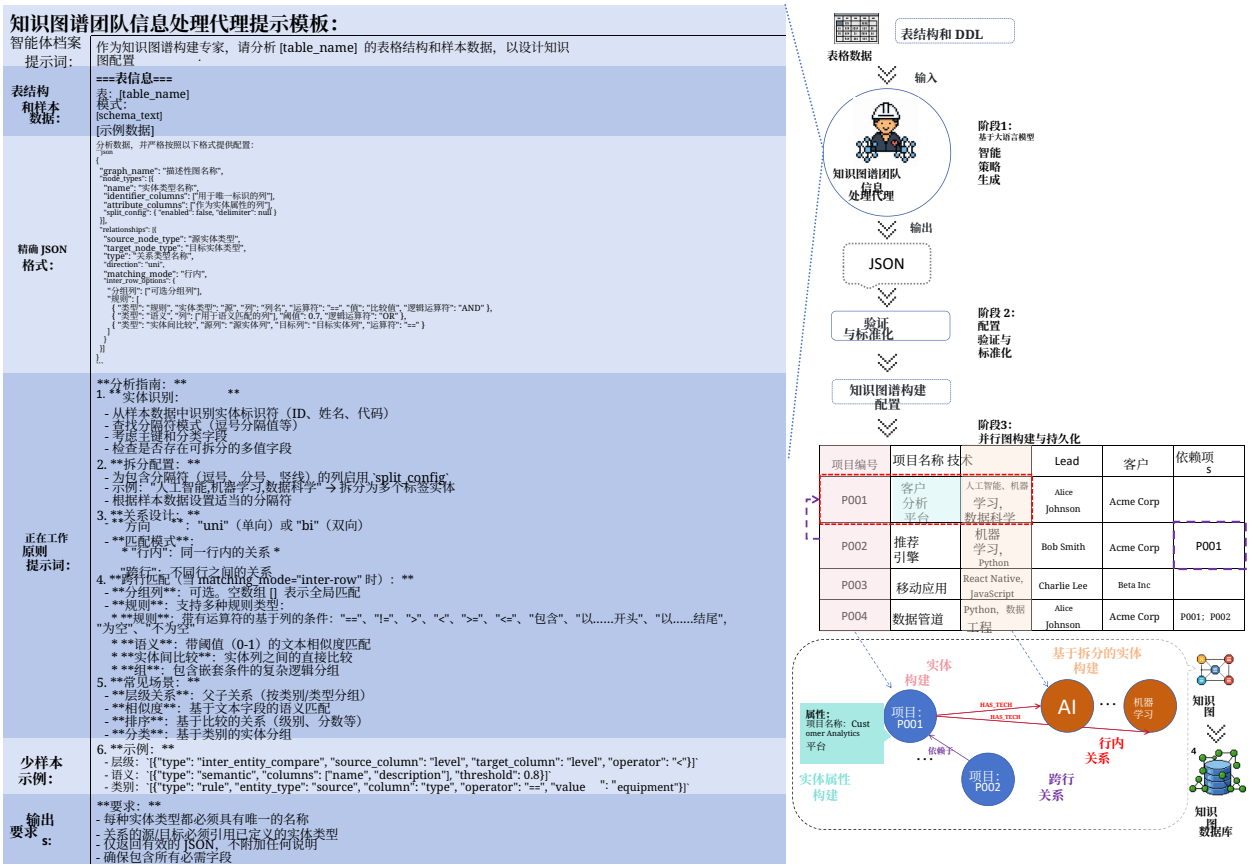


图5： 知识图谱信息处理智能体的工作流程。该智能体遵循策略生成、配置验证和图构建的三阶段流程，将表格数据转化为知识图谱。

通过系统验证实体定义和关系规则，确保算法一致性。图构建阶段实现形式化的实体识别和关系建立算法，并利用内存计算和批量持久化将数据存入 Neo4j 图数据库。图构建阶段的错误处理机制提供自动回退和状态恢复，确保系统在各种表格格式和数据质量场景下稳定运行。

4.3.2. 知识图谱信息检索智能体

知识图谱信息检索智能体实现了上下文增强的文本到Cypher生成，将结构化查询能力扩展至关系推理任务。该智能体通过实现与图结构的自然语言交互，解决了传统方法在处理多跳查询和语义关系方面的局限。

该智能体的核心创新在于面向图查询生成的自适应上下文构建。与静态模式方法不同，我们的方法从图形元数据、历史查询模式和领域知识中动态组装上下文信息。动态图模式提取通过查询Neo4j元数据接口获取当前节点标签、关系类型和属性键，为大语言模型提供对可用图结构和遍历可能性的全面理解。

上下文增强提示整合了三个组件，共同提升查询准确性。历史查询检索通过语义相似度匹配，从经验证示例中识别相关的问题-Cypher语句对，并根据结构兼容性筛选结果，其中包括节点标签重叠和关系模式。系统通过在每次交互后自动保存每条用户查询、其生成的 Cypher语句及执行结果，持续扩充历史问答数据存储库；所有条目均以向量化嵌入的形式存储在向量数据库中，从而能够高效地基于语义相似度进行检索，以增强未来的查询。



图 6： 知识图谱信息检索代理的架构。该智能体结合图模式信息、检索到的历史问题-Cypher 查询对以及外部领域知识生成 Cypher 查询，以支持稳健的多跳图遍历。

领域知识集成提供语义上下文，使大语言模型能够理解特定应用领域中的实体含义和关系语义。外部知识注入有助于生成融入领域特定推理模式的查询，超越基础图遍历的范畴。

该方法通过图遍历规划实现多跳推理，智能体会分析复杂查询中的关系要求，并构建结合实体属性检索与跨多个图级别的关系模式匹配的路径。此能力支持路径发现、模式匹配和关系推理等超出传统数据库操作范围的查询（见图 6）。

4.3.3. 知识图谱信息分析智能体与可视化智能体

知识图谱信息分析代理会解读图查询结果，并将结构化数据转化为自然语言响应。该代理通过对 Cypher 查询返回的节点、关系和路径进行语义分析，解决基于图的推理中的可解释性难题。

该代理通过系统化解读多种结果格式（包括节点集合、关系模式和多跳路径），识别查询上下文中的关键实体和关系。对于复杂的基于路径的结果，该代理会说明特定图遍历如何得出结论，从而解释推理链，使用户能够理解并验证多跳推断。这种透明性机制将技术性的图操作转化为易于理解的分析性解释。

结果集成会将图查询结果与用户意图相结合，生成连贯的响应，总结发现并提供相关背景信息。当涉及多个信息来源时，该代理会执行一致性验证，确保分析结论基于可靠的信息集成。自然语言生成会生成响应，解释直接答案以及从图关系中得出的支持性证据。

集成的可视化智能体通过生成交互式知识图谱可视化，在视觉呈现有助于理解时为分析流程提供补充。该智能体会评估查询结果，以确定合适的可视化方法，并通过网页界面创建子图显示，突出相关实体及其关系。此可视化能力使用户能够通过交互式图操作探索复杂的关系网络，并调查分析结果（见图 7）。

知识图谱团队分析智能体提示模板：

智能体档案 提示词：	你是一位智能助手，能够根据以下内容提供全面且有帮助的回答 图数据库查询结果
工作 原则 提示词：	**指令：** - 使用提供的信息构建清晰、详尽的答案 - 提供的信息具有权威性——绝不要质疑它，也不要用内部知识加以纠正 - 如果数据有限或不完整，请说明这一点，并解释现有的信息 - 未找到精确匹配时，请说明检索到了哪些相关信息 - 让回答听起来自然，口语化 - 始终如实说明数据局限性或缺失之处
少样本 示例：	**示例格式：** 问题：谁获得了游泳项目的金牌？ 上下文：[游泳运动员：John Doe，成绩：2:03.45，排名：1] 答案：根据现有数据，John Doe 以 2:03.45 的成绩获胜，排名第 1。 问题：专辑 X 在欧洲何时发布？ 上下文：[无结果] 答案：我找不到专辑 X 在欧洲的发布日期的具体信息。 当前数据集。 问题：谁是比赛中最后完赛的选手？ 上下文：[游泳运动员：Jane Smith，名次：4，时间：2:07。26]，[游泳运动员：Bob Jones，名次：5，时间：2:08。15] 回答：根据现有的比赛结果，最后记录的完赛者是 Jane Smith（名次 4，时间 2:07.26）和 Bob Jones（名次 5，时间 2:08.15）。请注意，这代表现有数据，而且可能还有其他参赛者。
附加 上下文：	**相关文档：** [领域知识在此处]
已检索 子图：	**查询结果：** [上下文]
问题：	**用户问题：**{question} 请根据现有信息提供有帮助的回答。



图7：知识图谱信息分析与可视化智能体的架构。分析代理以自然语言解释检索到的子图，子图绘制智能体则为 Web界面渲染相应的图形可视化。

4.4. 数据负责人与多智能体协作

数据负责人通过基于自然语言的协商机制和增强的系统可扩展性，实现了先进的推理编排，以达成我们的第一、第三、第四和第五项研究目标。该组件体现了我们的核心方法创新，即在统一框架内协调异构推理模式，从而能够在复杂推理过程中灵活调整策略并实现知识共享。

数据负责人的架构创新在于，通过 ReAct（推理与行动）范式，将传统僵化的工作流执行转变为动态协作智能。与依赖预先确定任务分配的传统多智能体系统不同，我们的方法使专业团队之间能够进行自然语言咨询与审议，从而根据查询复杂度及中间发现实时调整策略。该设计直接解决了现有方法在处理复杂多跳推理场景时的局限，同时通过显式协调机制保持可解释性。

该框架建立了三方协调架构，通过整合专业专长来分解复杂的表格问答任务。数据负责人采用系统化的“探索-验证-分析”三阶段原则来统筹协作式问题解决，将复杂推理任务转化为可管理的迭代循环，从而合并思考、动作片与观察。该方法能够实现动态规划、执行、反思与调整能力，有效应对多样化的查询要求与复杂推理路径（见图8）。

4.4.1. 三阶段原则

数据负责人采用系统化的三阶段分解策略，这是查询处理方法上的一项根本性创新。该方法将传统的查询优先执行模式转变为以证据为驱动的分析工作流，其灵感源自专业数据分析师面对未知数据集时所采用的系统化方法。该策略要求在构造查询之前先系统地探索数据，从而解决复杂表格问答场景中盲目查询这一关键局限。

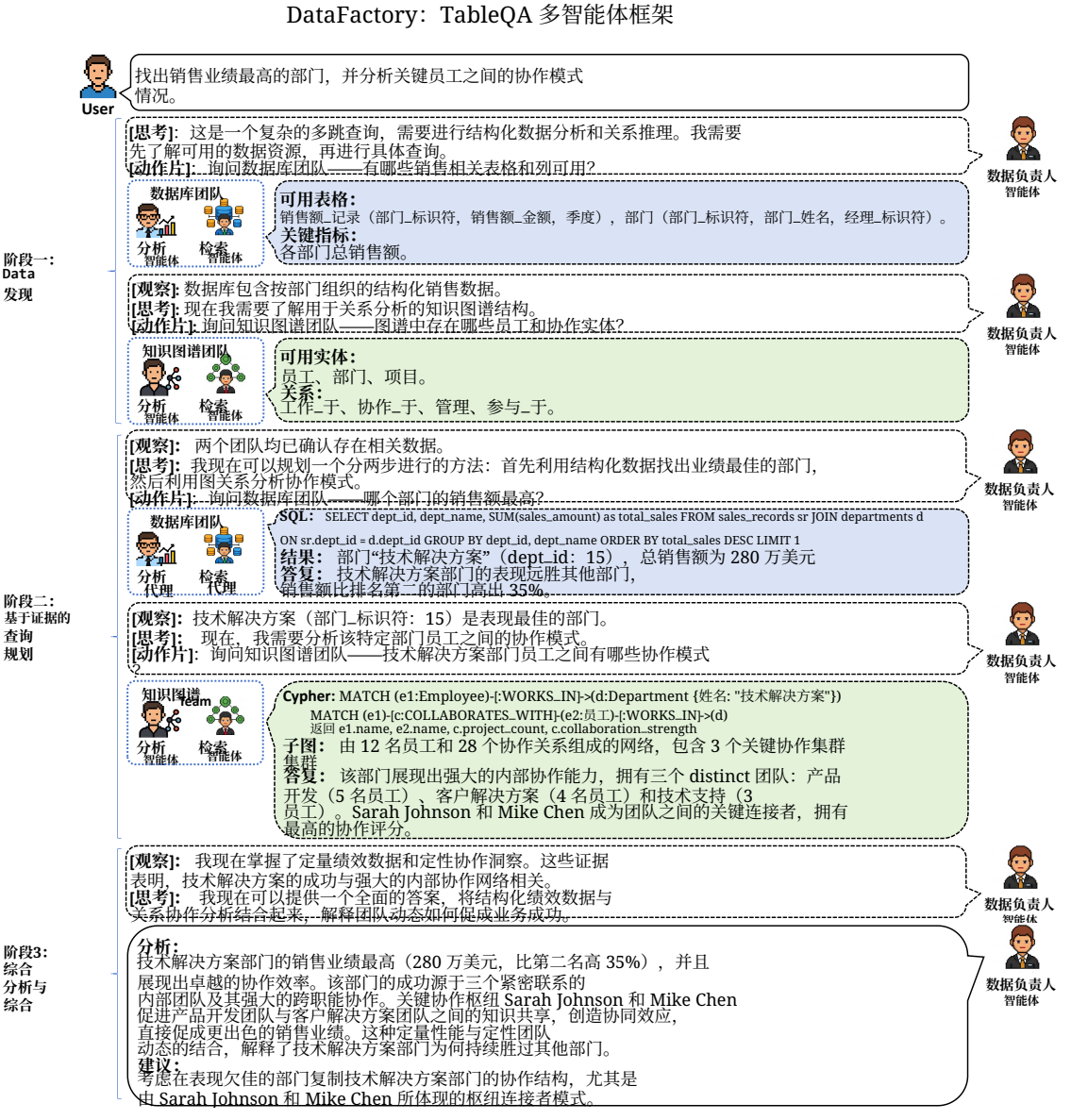


图8：数据负责人的工作流程。三阶段流程——数据发现、基于证据的查询规划以及综合分析与综合——实现ReAct范式，以协调数据库团队与知识图谱团队。

我们的方法首先强制执行数据发现：数据负责人会探索可用的数据资源，而不是直接根据用户要求进行查询。系统会系统地检查数据库表的结构信息，以了解可用字段和数据类型，同时探索知识图谱中已有的实体类型和关系类型。这种先探索的方法可避免盲目查询导致的问题，例如针对无效关键词或不存在的查询主题进行查询，从而为后续分析流程奠定可靠基础。

基于证据的查询规划阶段通过系统化的数据抽样与验证，扩展了传统的查询构造。数据负责人会抽样特定数据值，以确认用户感兴趣的实体或概念是否确实存在于数据中，然后再制定具体的查询策略。当直接查询无法获得预期结果时，系统会运用领域知识推理提出替代分析方案，从而显著提高查询成功率，同时避免无效操作造成的浪费。这种自适应方法使系统能够在特定实体缺失的情况下，通过识别相关行业数据或情境事件记录来处理相应场景。

全面分析与综合阶段通过精密的多步推理，整合来自多个来源的信息。数据负责人协调数据库查询和知识图谱探索所得的信息

，并从多个互补视角开展复杂的分析综合，以回应原始用户查询。此阶段通过系统整合结构性证据和关系性证据，将孤立的数据点转化为连贯的洞见。

对于含糊的用户请求，数据负责人会先通过简短的多轮自然语言交流进行澄清，以明确意图和约束，再进行规划；这种灵活的交互避免了僵化的输入/输出约定，并减少了需求描述不充分时的执行失败，同时通过保持连贯的对话状态，直接缓解 MAST 报告的规格定义错误和步骤重复问题（Pan 等人，2025）。

*4.4.2. ReAct*范式实现

数据负责人通过精心设计的迭代流程实现 ReAct 范式，该流程无缝整合了前文所述的三阶段原则（见图8）。这一实现体现了我们的核心创新：将复杂推理任务转化为易于管理的思考、动作和观察循环，从而实现动态适应能力和全面的问题解决能力，超越传统的多智能体协作方法。

我们的方法采用强大的大语言模型（LLM）作为中央推理引擎，并利用专门设计的系统提示词，使其具备超越单个智能体功能的元推理与协调能力（图9）。数据负责人通过自适应查询复杂度分析脱颖而出：将任务分解为易于管理的子任务，根据数据模态要求选择合适的团队，在检测冲突的同时综合多源信息，并在收集到充分证据后终止推理。分解策略采用自适应规划方法，由数据负责人根据查询复杂度确定推理步骤数，而非遵循预定规则，因此既能灵活处理只需少量迭代的简单事实查询，也能处理需要进行全面分析的复杂多跳推理。为防止无限推理循环并确保系统可靠性，用户可以配置最大推理步骤限制；超出预定义阈值时，超时机制会平稳终止推理，从而避免潜在的思维死锁。

动态团队调度机制是多智能体协作中的一项重要创新，它依赖语义意图分析，而非僵化的基于规则的委派。数据负责人采用基于能力的推理来确定最优团队选择：数据库团队擅长精确的结构化数据操作，包括数值计算、数据筛选、排序和大规模聚合；而知识图谱团队则擅长探索性实体关系任务，涉及路径发现、社区检测、关联分析和语义相似度推理。这种动态委派通过精细的上下文分析超越了基于关键词的路由，该分析会评估当前推理状态、可用证据以及查询推进要求。

观察与迭代反思机制通过对结果进行系统性评估，实现精细的自我纠正与复杂推理能力。数据负责人采用多准则评估，考察不同团队与推理步骤中的任务完成充分性、结果有效性以及信息集成要求。我们的创新包括一种数据驱动的冲突仲裁策略，通过数据溯源分析、一致性验证和领域知识集成来解决不同团队产生的矛盾结果。该方法使系统能够在保持分析严谨性的同时，提供承认多种视角的细致响应，从而将表格问答从孤立的工具调用转变为协同智能分析，以应对超出单个组件能力的复杂推理场景。

具体而言，冲突处理遵循评估-处理-验证协议：(i) 根据现有证据和溯源信息，判断结果是否充分、来源是否有效；(ii) 给出明确理由进行仲裁；(iii) 基于数据溯源，通过跨来源一致性检查进行验证；当数据库团队与知识图谱团队意见不一致时，系统会追溯溯源信息以确定分歧所在，而非任意信任任何一方的输出，从而解决 MAST 的“协调问题”和“规范违规”（Pan 等人，2025）。

4.4.3. 团队协作架构

团队协作架构体现了我们在统一框架内协调异构推理模式方面的基础性方法创新。这种方法将传统的层级关系或顺序协调模式转变为基于协商的协作模型，从而在复杂推理过程中实现灵活的团队间沟通和自适应策略调整。

数据负责人代理提示模板：



智能体档案 提示词：	你是一位系统化的数据分析专家 ·
工作 原则 提示词：	**系统化探索方法：** 阶段1 - 始终从数据发现开始： - 第一个动作片必须是发现存在哪些数据类型/列/实体 - 第二个动作片必须是对相关字段中的实际值进行抽样 - 只有完成发现之后，才能搜索特定实体 阶段 2 - 基于证据的查询： - 仅使用通过探索确认的值 - 根据发现的实际数据模式设计查询 阶段 3 - 全面分析： - 说明发现的数据局限性 - 如果目标实体不存在，则提出替代方案 **请记住：** 1. 首要动作必须始终是数据发现 2. 切勿假设特定实体存在 3. 严格遵循 Thought->Action->Action 输入格式 4. Action 输入应使用自然语言，而非 SQL/Cypher 5. 重要：当无法获取直接数据时，请运用领域知识来识别 实体类型并探索相关数据模式——例如，如果被问及“Tony Lema”以及 不存在球员数据，请识别出他是一名高尔夫球手，并查看锦标赛/高尔夫相关数据以进行 间接验证。 6. 关键：当你收到“表格上下文：该表格的数据与‘X’有关”时， 请理解整个表格都包含属于实体 X 的数据。不要在表格中搜索 X 的 姓名——相反，应将表格中的所有行视为 X 的数据记录。例如，如果表格 上下文提到“关于 'tony lema'”，而你看到锦标赛表现数据时，应将其理解为 Tony Lema 的个人锦标赛表现记录。 7.重要：如果两个团队得出的结论相互矛盾，始终应以实际 数据证据为最终依据。
ReAct 原则 提示词：	请严格按照以下格式。 **关键格式要求：** 1. 始终使用以下确切顺序：思考：-> 动作片：-> 动作片输入：-> 观察： 2. 思考：后绝不可省略“动作片：”这一行 3. 只能使用提供的确切工具名称 4. 每个循环必须完成后才能开始下一个 准确的工具名称：{sql_query, kg_query} 响应格式（强制）： 问题：[输入问题] 思考：[关于下一步该做什么的推理] 动作片：[以下工具名称中恰好一个：{tool_names}] 动作片输入：[针对该工具的自然语言问题] 观察：[来自该工具的结果] 思考：[对观察结果的分析以及后续步骤] 动作片：[如有需要，下一个工具名称] 动作片输入：[下一个工具的输入] 观察：[来自工具的结果] ...（继续，直到获得足够的信息） 思考：[最终分析，形成完整理解] 最终答案：[对原始问题的全面回答]
相关数据 上下文：	数据上下文： - SQL 数据库：'{sql_db}' - SQL 表：'{sql_table}' - 知识图谱：'{kg_ graph}' SQL 模式：{sql_schema_info} 知识图谱模式：{graph_schema_str}
问题：	开始！ 问题：{input} 思考：{agent_scratchpad}

图9： 数据负责人代理的提示模板。它编码了三阶段探索原则、两个团队的协调规则，以及推理过程中使用的 ReAct 风格思考–动作–观察格式。

我们的核心创新在于建立互补的专业领域，充分发挥结构化数据与关系数据处理范式的内在优势。数据库团队专注于数值精度、大规模聚合和确定性计算，而知识图谱团队则专注于语义推理、关系发现和多跳遍历。这种专业化使各团队能够针对各自的数据模式优化内部算法和提示策略，与试图统一处理所有查询类型的通用方法相比，取得更优的性能。为确保所有组件的性能一致，整个 DataFactory框架采用同一个大语言模型实例作为 DataLeader 和所有专业智能体的基础推理引擎，从而在多智能体协作过程中保持语言理解与生成能力的一致性。这项创新不仅限于简单的任务划分，还包括跨模态信息传输协议：数据库团队的定量结果作为知识图谱团队语义探索的输入约束，形成结构化到关系的信息流，在保持计算效率的同时实现深入复杂的推理。

数据负责人采用自适应任务委派策略，通过语义意图分析超越基于关键词的路由。委派流程纳入验证机制，通过可行性验证、依赖关系分析和一致性检查确保分解正确性。该验证框架

通过在整个推理过程中实施多个检查点，解决复杂查询场景中的可靠性问题。当发生验证失败时，数据负责人会启动纠正措施，包括备选团队选择、子任务重构或查询范围调整，从而在多样化的分析场景中实现稳健运行。

Data 负责人和专家团队之间的自然语言通信协议，代表了对结构化基于API的协调方式的重大转变。这一设计选择实现了丰富的上下文传递，其中不仅包括任务规范，还包括背景信息、目标导向和约束条件。自然语言的语义丰富性使团队能够在查询要求含糊或不完整时进行上下文推理。内置的容错机制包括语义澄清协议、优雅降级策略和替代建议能力，确保即使直接执行任务遇到障碍，也能持续运行。

这种协作架构将TableQA从孤立的工具调用转变为协同智能分析，确立了一种范式：通过结合专业专长、自适应委派和容错通信，处理超出单个组件能力范围的复杂推理场景。

实际上，清晰的角色边界、基于能力的调度以及内容丰富的自然语言消息，有助于防止角色/规范违规和对话历史丢失，而基于证据的停止则可避免终止无意识——这三种主要失败模式由MAST（Pan 等人，2025）强调。

5. 评估

为探究我们提出的结合结构与关系检索的多智能体表格问答方法的有效性，我们计划与该领域不同类型的方法进行性能比较，测试不同模型提供商及同一提供商不同模型规模下的相应性能，分析不同模型提供商和模型规模条件下 LeaderAgent 在不同团队中的调用模式，考察单团队检索与双团队协作对表格问答性能的影响，并探究团队协作交互频率与任务性能之间的关系。

我们提出以下 5 个研究问题（RQ）：

研究问题1： 我们的方法与其他方法相比表现如何，有哪些优势？我们将与五类典型方法进行比较，包括基于深度神经网络、基于提示工程优化、基于代码、基于智能体和基于多智能体的方法，并简要分析大语言模型令牌消耗量的模式。

RQ2： 不同模型提供商之间，以及同一提供商不同模型规模之间的性能有何差异？

RQ3： 在不同模型提供商和模型规模条件下，不同团队的领导智能体有哪些调用模式？它们之间存在何种差异？

RQ4： 与完整的多智能体框架相比，移除知识图谱团队会如何影响表格问答性能？

RQ5： 团队协作的交互频率与任务性能之间存在什么关系？达到最佳性能需要何种协作强度？

5.1. 实验设置

5.1.1. 基准测试

我们在三个表格问答基准数据集上评估了所提出的方法，这些数据集分别考察不同的任务视角：事实验证、计算分析和多跳推理。表1 总结了各数据集的主要特征和评估指标。

5.1.2. 大语言模型

我们使用来自5家提供商的8个大语言模型评估我们的框架，包括商业模型和开源模型。表2 总结了模型规格，其中3个商业模型需要API访问，5个开源模型支持本地部署。

5.1.3. 基线方法

我们从五大类别中选取具有代表性的方法进行比较：基于深度神经网络、基于提示工程优化、基于代码、基于智能体以及基于多智能体的方法：

表1基准数据集概览及其关键特征和指标

数据集	任务类型	测试集大小	表格大小	关键挑战	指标
TabFact (Chen 等人，2020)	事实验证（二元）	12,779 条陈述 1,695 个表格	5–6 列，14 行（平均值）	多跳推理，逻辑运算	准确率
WikiTQ （Pasupat 和 Liang，2015）	计算分析	4,344 个问题	≥8行，≥5 列	聚合、排序，算术运算	精确匹配准确率
FeTaQA (Nan 等人，2022 年)	多跳生成	2,003 个实例	5.9 列，13.8 行（平均值）	因果推理，自由形式答案	ROUGE-1/2/L F 分数

表 2用于评估的大语言模型及其关键规格

模型	提供方	Type	发布日期	参数	架构	部署
Claude 4.0 Sonnet （Anthropic，2024年）	Anthropic	商业	2025年5月	未披露（中量级）	稠密	API
Gemini 2.5 Flash (Google DeepMind,2024年)	谷歌 DeepMind	商业	2025年1月	未披露（轻量级）	稠密	API
GPT-4o mini (OpenAI，2024年)	OpenAI	商业	2024年7月	未披露（紧凑）	稠密	API
DeepSeek-V3 （DeepSeek-AI 等人，2025）	DeepSeek AI	开源	2024年12月	总计6710亿参数 激活370亿参数	MoE	API/本地
Qwen3-235B-A22B (Yang 等人, 2025a)	阿里巴巴云端	开源	2025年5月	总计2350亿参数 22B 激活参数	MoE	API/本地
Qwen3-32B (Yang等人，2025a)	阿里巴巴云端	开源	2025年5月	32B	稠密	API/本地
Qwen3-30B-A3B (Yang 等人，2025a)	阿里巴巴云端	开源	2025年5月	总参数30B 激活参数3B	MoE	API/本地
Qwen3-14B （Yang 等人，2025a）	阿里巴巴云端	开源	2025年5月	14B	稠密	API/本地

• **基于深度神经网络**：这些方法采用专为表格理解和问答设计的预训练深度神经网络模型。**TAPAS** (Herzig等，2020）扩展了 BERT 的架构，以联合编码自然语言问题和表格结构，从而能够通过表格单元格选择和聚合操作，基于弱监督进行端到端训练。**TAPEX** (Liu等，2022 年）通过在合成语料库上学习神经 SQL 执行器，引入了表格预训练，并通过自动合成可执行的 SQL 查询及其执行输出，应对数据稀缺问题。这两种方法都需要针对特定数据集进行训练，且跨领域泛化能力有限。我们采用在各自目标数据集上性能最优的大模型变体。

• **基于提示工程优化**：这些方法通过提示设计策略，引导大语言模型提升推理能力。**端到端**方法直接将表格和问题输入大语言模型以生成答案，而**少样本提示**则在提示词中加入1-5个示例，以提供学习参考。**TableCoT** （Ziqi和Lu，2023年）提出了一种表格格式的思维链（CoT），将表格问答分解为两步提示：首先引导模型生成表格格式的中间推理步骤，然后基于这些推理步骤提取最终答案，从而更好地利用表格的二维结构进行水平和垂直推理。

• **基于代码**：这些方法将表格内容转换为结构化数据，并使用代码（例如 SQL 语句或 Python 脚本）进行数据检索和分析。例如，**TabSQLify** （Pal 等人，2024年）提出通过文本到SQL生成将大型表格分解为更小的子表。这种方法可以显著减少大语言模型输入上下文，提升处理大型表格时的可扩展性和效率，并通过将子表选择与大语言模型推理相结合，生成准确的答案。

- **基于智能体**：这些方法赋予大语言模型工具使用、推理规划和记忆能力，使其能够更加自主地处理复杂的表格问答任务。例如，**StructGPT** （Jiang 等人，2023年）提出了一个通用框架，用于构建面向结构化数据推理的专用接口，并通过基于工作流的操作组织表格问答流程。然而，其数据操作接口主要依赖基于大语言模型的问答交互来执行行列提取等操作，而不是利用外部代码执行工具。 **ReAcTable** （Zhang 等人，2024b）提出了一种增强型 ReAct（推理与行动）范式方法，通过生成中间数据表示逐步增强输入数据。ReAcTable 将大语言模型推理能力与外部工具（例如 SQL查询执行器和 Python代码解释器）相结合，以执行复杂的数据操作和计算，从而有效处理表格问答任务。

- **基于多智能体**：近期研究探索了用于表格问答任务的多智能体框架，并为智能体明确分配专门角色。**MACT** (Zhou 等人，2025) 采用规划—执行多智能体工作流，由规划器使用 ReAct 范式制定数据操作计划，而代码智能体编写 Python 代码并调用计算工具，以执行数据操作与分析。**AutoPrep** (Fan 等人，2025) 进一步扩展了这一方法，在执行层为不同类型的数据操作（如筛选和标准化）设计专门的智能体，并分别使用 SQL 和 Python 语句执行非计算型和计算型操作。这些系统涉及大量交互工作流，其多智能体协作主要依赖函数式参数交互，而非自然语言咨询机制。

实验配置与参数设置。为确保所有基于大语言模型的方法之间能够公平比较，我们采用相同的基于API的配置，并将温度设为=0.1，以确保结果可复现。请注意，基于深度神经网络的方法（TAPAS 和 TAPEX）由于架构存在根本差异，沿用其原始配置。少样本、TableCoT、StructGPT、ReAcTable、MACT、AutoPrep 和我们的 DataFactory 均使用 3 个示例。ReAcTable、MACT、AutoPrep 和我们的数据负责人均将交互轮次上限设为 20。所有评估均使用相同的大语言模型（GPT-4o mini、Deepseek-V3、Gemini 2.5 Flash），并采用一致的 API 设置。

5.2. 研究问题1：与其他方法的比较

本节将我们的方法与第5.1.3节提出的五类共十种基线方法进行比较，并使用第5.1.1 节介绍的前两个数据集进行评估。我们聚焦于这两个数据集，因为大多数基线方法尚未在 FeTaQA 上进行测试，也没有提供相应的实现代码或实验结果。在评估与大语言模型相关的方法时，我们采用了三种主流模型：GPT-4o mini、Deepseek-V3 和 Gemini 2.5 Flash。考虑到方法稳定性，我们在每个数据集上对每种方法（涉及大语言模型时，使用相应的大语言模型）运行三次，计算每种方法-模型-数据集组合的均值和方差，并计算所有模型和运行结果的总体均值，作为代表性性能指标。为了评估性能波动下改进的统计显著性，我们计算同一数据集上使用同一大语言模型的对应方法的 Cohen's d 值（Cohen，1988），评估配对比较中的效应量，并计算三个模型的 Cohen's d 均值，以说明在性能波动范围内我们的方法所取得的性能提升的显著性。详细实验结果见表3。

从方法论角度看，这两种基于深度神经网络的方法在 TabFact 上取得了良好性能，但在 WikiTableQuestions 上表现较差，反映出此类方法在复杂表格检索和计算任务上的局限性。此外，每个数据集都需要单独训练以形成相应模型，这不利于实际应用，也无法适应复杂的表格类型和问答场景。端到端、少样本和 TableCoT 方法都将表格数据作为上下文输入到大语言模型中。这些方法在 TabFact 二元分类任务上表现尚可，但在 WikiTQ 检索和计算

[3]https://github.com/google-research/tapas[4]
https://github.com/microsoft/Table-Pretraining[5]
https://github.com/TIGER-AI-Lab/TableCoT[6]
https://github.com/RUCAIBox/StructGPT[7]
https://github.com/mahadi-nahid/TabSQLify[8]
https://github.com/yunjiazhang/ReAcTable[9]
https://github.com/boschresearch/MACT[10]
https://github.com/ruc-datalab/AutoPrep

表3不同方法在 TabFact 和 WikiTQ 数据集上的性能比较

方法	TabFact					WikiTQ				
	结果			平均值 (改进)	Cohen 的 d	结果			平均值 (提升)	Cohen 的 d
TAPAS-large[3]	80.1±1.2%			80.1% (↑3.9%)	3.25	50.7±2.8%			50.7% (↑22.1%)	7.89
TAPEX-large[4]	82.0±1.5%			82.0% (↑2.9%)	1.33	56.2±1.7%			56.2% (↑16.6%)	9.76
	GPT-4o mini	Deepseek V3	Gemini 2.5 Flash			GPT-4o mini	Deepseek V3	Gemini 2.5 Flash		
端到端[5]	69.4±2.1%	76.0±1.0%	75.3±1.5%	73.6% (↑14.1%)	7.27	50.7±0.6%	55.9±0.3%	57.1±1.1%	54.6% (↑33.3%)	16.59
少样本[5]	64.2±1.7%	79.2±2.0%	63.6±0.8%	69.0% (↑21.7%)	16.03	42.7±1.2%	55.6±2.5%	50.6±2.9%	49.6% (↑46.7%)	13.94
TableCoT[5]	51.0±1.1%	79.8±6.4%	50.6±2.3%	60.5% (↑38.9%)	16.99	32.3±2.2%	60.6±4.6%	42.2±2.4%	45.0% (↑61.7%)	13.65
StructGPT[6]	58.4±0.8%	55.6±4.6%	64.4±3.6%	59.4% (↑41.3%)	13.53	49.8±1.4%	65.5±1.5%	70.9±0.9%	62.1% (↑17.2%)	8.03
TabSQLify[7]	78.3±1.3%	74.5±4.2%	73.9±1.1%	75.6% (↑11.1%)	6.9	**74.0±1.4%**	68.0±3.2%	66.7±0.5%	69.6% (↑4.6%)	5.24
ReAcTable[8]	69.4±1.9%	68.1±5.1%	74.0±0.9%	70.5% (↑19.1%)	9.9	57.8±3.1%	56.8±2.9%	70.4±3.6%	61.7% (↑18.0%)	4.61
MACT[9]	47.4±1.3%	58.9±1.2%	72.9±5.2%	59.7% (↑40.6%)	14.47	62.3±0.2%	68.6±3.0%	67.3±2.3%	66.1% (↑10.1%)	4.8
AutoPrep[10]	74.6±2.1%	80.7±3.5%	76.6±3.6%	77.3% (↑8.6%)	2.96	65.9±3.3%	67.5±4.1%	68.2±1.8%	67.2% (↑8.3%)	2.57
我们的方法(DataFactory)	**80.1±1.0%**	**82.2±2.9%**	**89.6±0.7%**	**84.0% (↑20.2%)**	**9.26**	67.7±1.0%	**73.5±1.7%**	**77.2±1.7%**	**72.8% (↑23.9%)**	**8.71**

任务上表现较差。然而，与前一类别相比，它们在泛化能力方面显著受益于大语言模型的引入。相比之下，其余方法均具备数据操作与深度分析能力，从而使 WikiTQ 数据集上的性能提升到了新的级别。

在基于大语言模型的方法中，基于上下文的方法受到大语言模型上下文长度限制和模型幻觉的制约，无法处理过大的表格数据，也难以为复杂检索和计算任务提供准确答案。此外，当表格问答完全依赖上下文时，过多的上下文信息和冗长复杂的提示词会对 GPT-4o mini 和 Gemini 2.5 Flash 产生不利影响，导致性能下降，同时加剧性能波动。因此，这三种方法在这两个数据集上的性能和稳定性均呈下降趋势，这表明在表格问答中继续探索新的上下文依赖型提示模式和方法未必总是有益。

其余方法都具备数据操作与交互能力，而我们的方法在大多数模型上均优于其他方法。这类方法通常受到大语言模型能力的限制，需要模型具备足够的智能，能够理解指定的解决策略并提供合理的解决步骤。StructGPT 可以通过大语言模型问答交互，针对特定问题提取并分析行和列的内容，但基于大语言模型的数据提取在准确率和可靠性方面仍然有限，容易受到大模型幻觉影响，并受限于大语言模型的数据提取能力。与我们的方法相比，TabSQLify 方法在两个数据集上的表现都较差，因为它只考虑了表格数据的结构理解，没有涉及关系检索。尽管 ReAcTable 引入了 ReAct 机制，但其单智能体设计要求智能体同时处理 SQL 语句生成与数据检索、检索结果的内容分析，以及整体解决策略的制定。这种过重的任务负担导致其整体性能不如若干基线方法。具有代表性的多智能体基线方法通常依赖广泛的工作流编排和代码执行，这增加了交互复杂度和错误面。相比之下，我们的框架强调清晰的角色专门化，以及以声明式 SQL/Cypher 执行为核心的自然语言协调，从而实现了更高的准确率和更强的稳定性。

在最终性能方面，我们的方法优于基于 DNN 的传统方法，并在全部三个模型上超过其他基于 LLM 的方法，尤其在 WikiTQ 上表现突出。在 TabFact 数据集上，与其他方法相比，我们的方法平均提升了 20.2%；而在 WikiTQ 数据集上，平均提升了 23.9%。Cohen's d 值均大于 1，表明改进效果显著，展现出在表格问答任务中更强的适应性和性能。

关于计算效率，我们分析了所有模型和运行中基于大语言模型的方法的平均令牌消耗量（输入 + 输出），详情见表4。结果显示，基于提示的方法< 基于代码的方法 < 单智能体方法 < 多智能体方法。在多智能体方法中，我们的设计采用更少的组件，并简化数据提取流程（仅使用 SQL 和 Cypher），因此令牌消耗量相对较低。虽然与其他方法类别相比，令牌消耗量较高，但考虑到性能提升，我们认为这项成本是必要的；清晰、透明的分析流程也增强了用户对问答过程的信任。

表4不同方法的平均令牌使用量比较

方法	TabFact 平均词元数	WikiTQ 平均 Token 数
端到端	546	794
少样本	935	1,189
TableCoT	1,084	1,495
StructGPT	1,721	1,999
TabSQLify	2,194	2,317
ReAcTable	3,050	4,400
MACT	3,673	5,319
AutoPrep	4,013	5,870
我们的方法（DataFactory）	**3,464**	**4,982**

表5不同模型提供商和规模的性能比较

模型	FeTaQA			WikiTQ	TabFact
	ROUGE-1 F	ROUGE-2 F	ROUGE-L F	准确率	准确率
Claude 4.0 Sonnet	**0.6234**	**0.3885**	**0.5067**	**83.2%**	90.8%
Gemini 2.5 Flash	0.6107	0.3767	0.4995	77.9%	87.9%
Qwen3-235B-A22B	0.5972	0.3607	0.4776	73.2%	**91.2%**
DeepSeek-V3	0.5403	0.3075	0.4309	73.6%	84.2%
GPT-4o mini	0.5576	0.3320	0.4555	67.8%	79.3%
Qwen3-32B	0.5662	0.3327	0.4569	61.0%	79.8%
Qwen3-30B-A3B	0.5063	0.2899	0.4091	48.1%	75.0%
Qwen3-14B	0.5289	0.3033	0.4253	53.7%	75.7%

研究问题1总结回答

我们的多智能体框架通过专业化的团队协作，在两个主要数据集上的表现显著优于基线方法：数据负责人负责策略制定，数据库团队负责结构化 SQL 操作，知识图谱团队负责关系推理。性能提升幅度显著——在 TabFact 上提升 20.2%，在 WikiTQ 上提升 23.9%，Cohen's d 为 > 1，表明结果具有统计显著性。尽管令牌成本较高，但清晰的分工和自然语言交互使其比其他方法更稳定、可解释性更强。

5.3. RQ2：不同模型提供方和规模下的性能

为了探究不同模型提供商之间，以及同一提供商旗下不同规模模型之间的性能差异，我们使用来自五家提供商的八个大语言模型，在三个基准数据集上进行了系统性实验。实验结果如表 5所示，揭示了模型特征与表格问答任务性能之间的若干重要关系。

实验结果表明，不同模型之间以及同一模型系列内部存在明显的性能层级。能力更强的模型得分最高，其中 Claude 4.0 Sonnet 在所有指标上均领先：在 FeTaQA 上的 ROUGE-1 F 分数为 0.6234，在 WikiTQ 上的准确率为 83.2%，在 TabFact 上的准确率为 90.8%，展现出处理复杂表格问答任务的强大能力。

在复杂推理任务上，性能差距进一步扩大。Gemini 2.5 Flash 表现稳健，而 Qwen3-235B-A22B 在事实验证任务（TabFact）上以 91.20% 的准确率超越 Claude 4.0 Sonnet，这表明大规模模型在结构化推理方面极具竞争力，不过部分模型在需要高质量自由形式生成的任务（如 FeTaQA）上仍略显逊色。

在Qwen3系列中，性能与模型规模和架构呈现出明显的递增关系：Qwen3-235B-A22B ＞Qwen3-32B ＞ Qwen3-30B-A3B ＞Qwen3-14B。值得注意的是，尽管总参数量相近，稠密架构的Qwen3-32B始终优于混合专家架构的Qwen3-30B-A3B，这表明稠密模型可能更适合依赖全面参数利用的复杂推理任务。

对不同任务复杂度的比较也揭示了模型性能下降的规律：TabFact作为二元分类任务，模型间的性能差距最小，所有模型的准确率均超过75%；WikiTQ要求精确匹配，差距适中；FeTaQA需要多跳推理和自由文本生成，差距最大，ROUGE-1 F分数介于0.5063至0.6234之间，凸显了表格问答场景中自由文本生成的挑战。

我们的多智能体数据工厂架构在不同模型能力下展现出良好的适应性。即使是 Qwen3-14B 这样规模较小的模型，在 TabFact 上也能达到 75.65% 的准确率，这表明专业化的团队协作和结构化的提示词设计可以部分弥补单个模型的局限性。数据库团队的 SQL 生成和知识图谱团队的 Cypher 查询受益于上下文增强提示范式，能够有效引导较小的模型生成正确的查询。此外，这些较小模型的性能也超过了表3中所示的基于上下文的表格输入方法，展现出我们方法的优势。

总体而言，该框架的有效性随模型能力提升而增强，同时在不同模型规模下保持稳健性。顶尖模型能够充分利用数据负责人提出的 ReAct 策略，实现最佳规划与协调；不过，规模较小或能力较弱的模型也能通过系统化的任务分解和智能体协作取得有竞争力的性能，为资源受限场景下的本地部署提供可行方案。

从可扩展性的角度来看，这些结果初步定量展现了 DataFactory 在两个实际维度上的表现。首先，通过采用来自五家提供商的八种模型实例化相同的多智能体设计，涵盖广泛的参数量范围（表5），我们观察到，随着模型容量降低，性能会平稳下降而非骤然崩溃，这表明该框架适用于不同的计算预算。其次，表4中的词元使用量比较与表3中的准确率提升共同表明，从基于提示的基线方法转向基于代码的方法和多智能体方法时，准确率与成本之间大致呈单调权衡关系；我们的设计则处于一个额外交互成本能够换来明显性能收益的位置。

RQ2 总结性回答

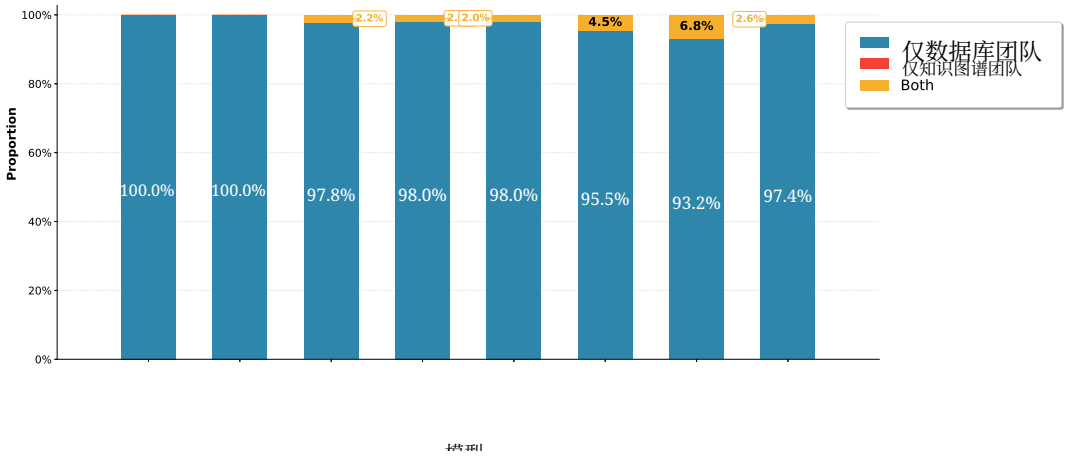
总体而言，能力更强的模型表现更优，其中 Claude 4.0 Sonnet 在大多数指标上取得了最佳性能。在同一模型系列中，性能会随模型规模扩大而提升（Qwen3-235B > Qwen3-32B > Qwen3-30B >Qwen3-14B），并且在这些任务中，稠密架构优于混合专家模型。我们的框架能够很好地适应不同规模，而较小的模型也能通过专门化的团队协作取得具有竞争力的结果。

5.4. RQ3：领导智能体团队调用模式分析

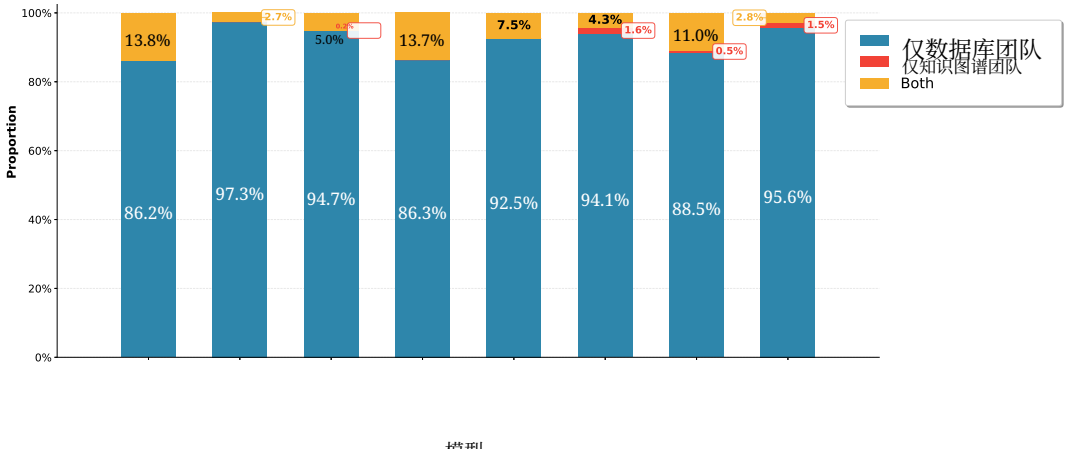
为分析领导智能体如何在不同任务类型、模型提供商和模型规模下调用不同团队（数据库团队和知识图谱团队，或两者同时调用），我们在 RQ2 的基础上，对全部三个基准数据集开展了全面的工具使用模式分析。本研究揭示了任务特征和模型能力如何影响智能体的协作策略与工具选择偏好。

我们的分析表明，在三个基准数据集中，不同模型调用数据库团队（基于SQL的检索）和知识图谱团队（基于 Cypher 的检索）的方式存在显著差异。详细分析揭示了基于任务复杂度和模型特征的不同模式（见图 10）。

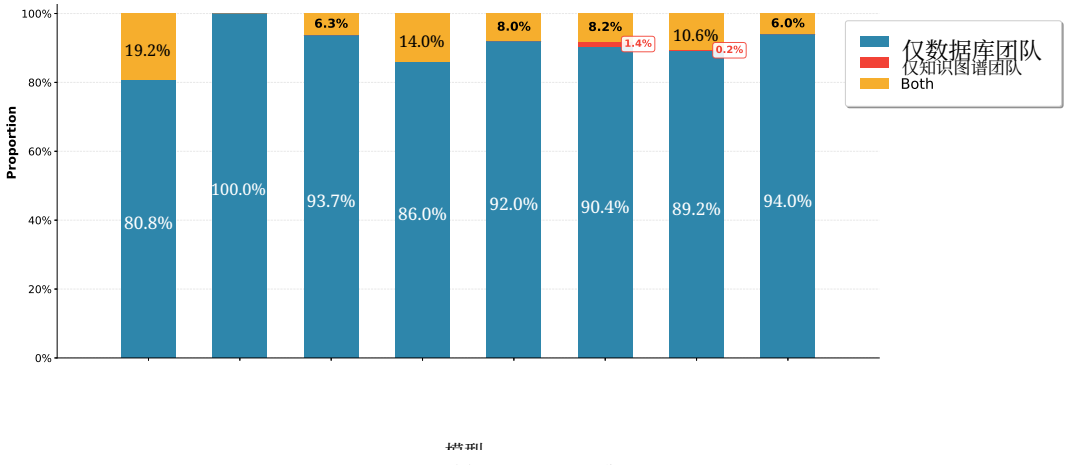
首先，关于数据集差异，数据负责人在不同模型上的查询策略在 TabFact 数据集上最为一致，主要选择以数据库团队为中心的查询方式，极少或完全不调用知识图谱团队。这是因为 TabFact 任务主要涉及二元分类验证，而事实验证的二元特性更适合直接进行 SQL 事实核查；此外，表3所示不同方法普遍较高的准确率也表明，任务难度相对较低。相比之下，在 WikiTQ 和 FeTaQA 数据集上，模型的查询策略更加多样，知识图谱团队的调用率显著上升，且大多数模型采用同时调用数据库团队和知识图谱团队的策略。具体而言，WikiTQ 任务需要复杂检索、匹配和计算分析，策略多样性适中，同时更多地使用知识图谱团队进行关系推理。在 FeTaQA 上，任务复杂度和多跳推理特性使模型更倾向于结合使用两个团队的工具，合用率最高可达 19.2%（Claude 4.0 Sonnet），这表明此类自由形式问答问题更加依赖多源信息收集和全面分析。



（a）TabFact 数据集



（b）WikiTQ 数据集



（c）FeTaQA 数据集

图 10： 不同模型和数据集上的团队调用模式。该可视化展示了任务复杂度如何影响工具选择策略：（a）由于具有二元分类的特性，TabFact 一致倾向于使用数据库团队；（b）WikiTQ 的工具选择呈现中等程度的多样化，在复杂检索任务中更频繁地使用知识图谱；（c）FeTaQA 的团队联合使用率最高，反映了其多跳推理要求。蓝色表示仅使用数据库团队，红色表示仅使用知识图谱团队，橙色表示联合使用两个团队。

在全部三个数据集上，所有模型都更倾向于采用基于SQL的数据库检索，而各模型对知识图谱的使用仍相对有限。我们认为，这是因为现有表格问答任务主要考查结构化查询，而非关系推理；许多任务都可以直接通过 SQL 解决。然而，知识图谱团队调用频率较低，并不意味着它能力不足或效果不佳——相反，这体现了基于图的推理具有互补性和专业性。正如 RQ4 分析（第 5.5 节）所示，即使知识图谱团队的使用似乎有限，其贡献仍能带来稳定的性能提升：TabFact 上提升 2.6%，WikiTQ 上提升 5.5%，FeTaQA 的 ROUGE-2 指标最高提升 7.8%。知识图谱团队的战略价值不在于使用频率，而在于它能够精准处理需要语义关系和多跳逻辑的特定推理场景。在实际应用中，知识图谱的使用仍是一项重要补充，尤其适用于超出 SQL 结构化查询能力的复杂关系分析。

关于不同模型的能力，可以观察到以下特点：Claude 4.0 Sonnet 在 FeTaQA 上的综合使用率最高（19.2%），表明其在协调多个团队方面具备出色的规划能力；其次是 DeepSeek-V3，这两个模型都展现出多样化的调用策略。Gemini 2.5 Flash 最为保守，在所有数据集上仅调用 SQL，说明它倾向于选择更简单、更可靠的问题求解路径。对比稠密架构与混合专家架构，混合专家模型 Qwen3-30B-A3B 在全部三个数据集上的工具综合使用率都更高，其团队平均综合使用率（9.47%）显著高于稠密模型 Qwen3-32B（5.67%），这表明混合专家架构可能会采用更具探索性的检索策略。

RQ3 总结性回答

任务复杂度显著影响团队调用模式：TabFact 倾向于单独使用数据库团队，而 FeTaQA 的团队综合使用率最高（Claude 最高达 19.2%）。总体而言，所有模型都更倾向于基于 SQL 的检索，但 Claude 4.0 Sonnet 等先进模型展现出更强的多团队协作能力。与稠密模型相比，MoE架构展现出更具探索性的策略，工具组合使用率也更高。知识图谱团队调用频率较低，反映的是由任务驱动的专业化，而非能力不足——在实际部署中，其贡献主要体现在以数据关联和语义关系为核心的任务上，并能提供独特的推理能力。

5.5. RQ4：移除知识图谱团队对表格问答性能的影响

为评估知识图谱团队的必要性和有效性，我们开展了全面的消融实验，在全部八种模型和三个基准数据集上，对比移除知识图谱团队后的性能与完整多智能体框架的性能。表 6 系统地比较了不包含知识图谱团队的框架配置（仅使用结构化 SQL 查询）与集成 SQL 和基于 Cypher 的检索方法的完整框架。

全面的消融研究表明，集成知识图谱团队能够持续提升所有数据集上的性能，在多跳推理任务上尤为显著。性能提升随任务复杂度系统性地增长：TabFact 的提升适中（平均 5.5%），因为二元事实验证主要需要结构化 SQL 查询；而 WikiTQ（平均提升 14.4%）和 FeTaQA（各项指标提升 12.8%–17.1%）则展现了关系推理在复杂检索和多跳推理任务中的重要价值。WikiTQ 和 FeTaQA 都是典型的多跳问答数据集，需要理解语义关系并进行跨实体推理，而传统的纯 SQL 方法无法充分应对这些需求。

模型能力层级对多智能体协作表现出不同的响应。较小参数量的模型（Qwen3-32B、Qwen3-14B）在双团队协作下提升最为显著：在 WikiTQ 上的准确率分别提高了 44.9% 和 37.7%，在 FeTaQA 上的 ROUGE-2 分数则分别提升了 52.6% 和 48.5%。在复杂多跳数据集上取得的这些显著提升表明，知识图谱团队的整合能够有效弥补较小模型在表格问答中的局限，增强其对复杂关系和语义结构的理解与推理能力。相比之下，参数量较大的模型基线性能更稳定，协作带来的收益虽持续但较为适中，这反映了不同架构对多智能体协作的适应性差异。

知识图谱团队通过 Cypher 查询提供的关系推理能力，系统性地解决了纯 SQL 方法的根本局限。在需要跨行关系

表6无知识图谱团队的框架与完整框架的性能对比

数据集	指标	模型	无知识图谱团队	完整框架	平均提升
TabFact	准确率	Claude 4.0 Sonnet	86.7%	90.8% (↑4.7%)	5.5%
		Gemini-2.5 Flash	85.1%	87.9% (↑3.3%)	
		Qwen3-235B-A22B	85.6%	91.2% (↑6.5%)	
		DeepSeek Chat	82.0%	84.2% (↑2.7%)	
		GPT-4o mini	77.3%	79.3% (↑2.6%)	
		Qwen3-32B	67.7%	79.8% (↑17.9%)	
		Qwen3-30B-A3B	73.0%	75.0% (↑2.7%)	
		Qwen3-14B	73.0%	75.7% (↑3.7%)	
WikiTQ	准确率	Claude 4.0 Sonnet	78.2%	83.2% (↑6.4%)	14.4%
		Gemini-2.5 Flash	74.0%	77.9% (↑5.3%)	
		Qwen3-235B-A22B	70.0%	73.2% (↑4.6%)	
		DeepSeek Chat	69.0%	73.6% (↑6.7%)	
		GPT-4o mini	64.9%	67.8% (↑4.4%)	
		Qwen3-32B	42.1%	61.0% (↑44.9%)	
		Qwen3-30B-A3B	45.8%	48.1% (↑5.0%)	
		Qwen3-14B	39.0%	53.7% (↑37.7%)	
FeTaQA	ROUGE-1 F	Claude 4.0 Sonnet	0.5963	0.6234 (↑4.5%)	12.8%
		Gemini-2.5 Flash	0.5829	0.6107 (↑4.8%)	
		Qwen3-235B-A22B	0.552	0.5972 (↑8.2%)	
		DeepSeek Chat	0.5161	0.5403 (↑4.7%)	
		GPT-4o mini	0.5173	0.5576 (↑7.8%)	
		Qwen3-32B	0.4209	0.5662 (↑34.5%)	
		Qwen3-30B-A3B	0.4868	0.5063 (↑4.0%)	
		Qwen3-14B	0.3954	0.5289 (↑33.8%)	
	ROUGE-2 F	Claude 4.0 Sonnet	0.3675	0.3885 (↑5.7%)	17.1%
		Gemini-2.5 Flash	0.3634	0.3767 (↑3.7%)	
		Qwen3-235B-A22B	0.3365	0.3607 (↑7.2%)	
		DeepSeek Chat	0.2975	0.3075 (↑3.4%)	
		GPT-4o mini	0.296	0.332 (↑12.2%)	
		Qwen3-32B	0.218	0.3327 (↑52.6%)	
		Qwen3-30B-A3B	0.2794	0.2899 (↑3.8%)	
		Qwen3-14B	0.2042	0.3033 (↑48.5%)	
	ROUGE-L F	Claude 4.0 Sonnet	0.4863	0.5067 (↑4.2%)	12.4%
		Gemini-2.5 Flash	0.4852	0.4995 (↑2.9%)	
		Qwen3-235B-A22B	0.4522	0.4776 (↑5.6%)	
		DeepSeek Chat	0.4193	0.4309 (↑2.8%)	
		GPT-4o mini	0.4153	0.4555 (↑9.7%)	
		Qwen3-32B	0.3394	0.4569 (↑34.6%)	
		Qwen3-30B-A3B	0.3907	0.4091 (↑4.7%)	
		Qwen3-14B	0.316	0.4253 (↑34.6%)	

分析、语义相似度判断和多跳推理路径的场景中，知识图谱团队的贡献变得不可或缺。这解释了为什么协作效应在 FeTaQA 类型的自由形式问答和 WikiTQ 风格的复杂检索任务中最为显著；这两类任务都体现了我们的框架专门设计用于解决的多跳推理挑战。自动化知识图谱构建能够发现传统结构化查询方法无法显现的隐式实体关系和语义联系，从而支持高级表格问答场景所需的复杂推理模式。

RQ4 总结性回答

在所有数据集上，完整框架的表现始终优于移除知识图谱团队的配置，提升幅度从 5.5%（TabFact）到 17.1%（FeTaQA ROUGE-2）不等。针对八种模型进行的全面消融研究表明，移除知识图谱团队会显著降低性能，尤其会影响具有挑战性的多跳问答场景所需的语义关系分析和跨实体推理能力。性能损失会随任务复杂度系统性增加，这证明了知识图谱团队的关系推理能力不可或缺，尤其是在 WikiTQ 和 FeTaQA 数据集上；这两个数据集都是典型的多跳推理基准，也是本框架专门针对的任务。

5.6. RQ5：团队协作频率对性能的影响

为了解协作频率如何影响性能，我们分析了 TabFact 和 WikiTQ 数据集上所有模型的团队交互频率与任务准确率之间的关系。我们将交互频率分为五个区间：单次交互（1 次调用）、低频（2–3 次调用）、中频（4–5 次调用）、高频（6–10 次调用）以及极高频（10+ 次调用）。我们的假设是，存在一个使性能达到最高的最佳交互频率，而交互频率过高或过低都会导致性能下降（见图11 和图 12）。

我们的分析揭示，协作频率与性能之间存在明显的倒U形关系，尽管任务复杂度不同，两个数据集都呈现出相似的模式。性能在低频交互范围（1–3 次调用）达到峰值，随后随着交互频率增加而逐渐下降。

TabFact 数据集在调用 1 次（85.4%）和 2–3 次（81.9%）时表现最佳，且标准差相对较小（±6.9% 和 ±8.3%），表明在简单的二元分类任务中，各模型的性能较为稳定。WikiTQ 在调用 4–5 次时性能达到峰值（87.1%），但之后下降更为显著，在 10+ 次调用时降至 20.2%，且方差增大（±22.5%）。高频交互下较大的误差棒反映出所测试的八个模型之间固有的性能差异，这些模型涵盖了不同的能力层级和架构。此外，在我们针对这两个数据集开展的实验中，高频交互场景（6+ 次调用）出现频率相对较低，因此数据点分布更为分散，如图 12b所示。

热力图分析揭示了模型能力和任务复杂度所对应的不同模式。在 TabFact（图 12a）中，大多数模型在调用 1–3 次时表现最佳；Claude 4.0 Sonnet 在调用 2–3 次时达到 94.3%，Qwen3-235B 则在调用 4–5 次时达到 100%。这种稳定性反映了验证任务对交互频率的容忍度。相较之下，WikiTQ（图 12b）表现出更明显的早期收敛：Claude 4.0 Sonnet 和 DeepSeek-V3 在单次调用时分别达到 100% 和 95.7%，但随着交互次数增加，性能迅速下降。

不同模型能力层级表现出不同的协作偏好。顶级模型（如 Claude 4.0 Sonnet、Gemini 2.5 Flash）在低频交互下表现出色，展现出更强的单轮规划能力。中等规模模型（Qwen3-32B、GPT-4o mini）则能从适度迭代（调用 2–4 次）中受益，而较小模型（Qwen3-14B）的表现波动较大：随着交互次数增加，其在 WikiTQ 上的准确率从 65.5% 降至 0%。

最显著的是，当调用次数超过 6 次时，我们观察到一致的下降趋势，模型在 WikiTQ 上的性能出现下降。GPT-4o mini 从 77.0%（2–3 次调用）下降至 5.9%（10+ 次调用），充分体现了这一模式，表明过度协作会导致决策混乱、错误累积以及推理链中断。尽管由于各模型能力不同，其在高交互频率下的性能表现存在显著差异，但总体退化趋势在不同模型类型中保持一致。

RQ5 总结答案

团队协作存在一个最佳交互频率，性能在 1–3 次调用时达到峰值，之后开始下降。顶级模型展现出更强的早期收敛能力，而过度协作（6+ 次调用）会导致性能严重退化，尤其是在复杂任务上。模型能力与最佳协作频率之间的关系，为在多智能体系统中设计自适应停止机制提供了重要见解。

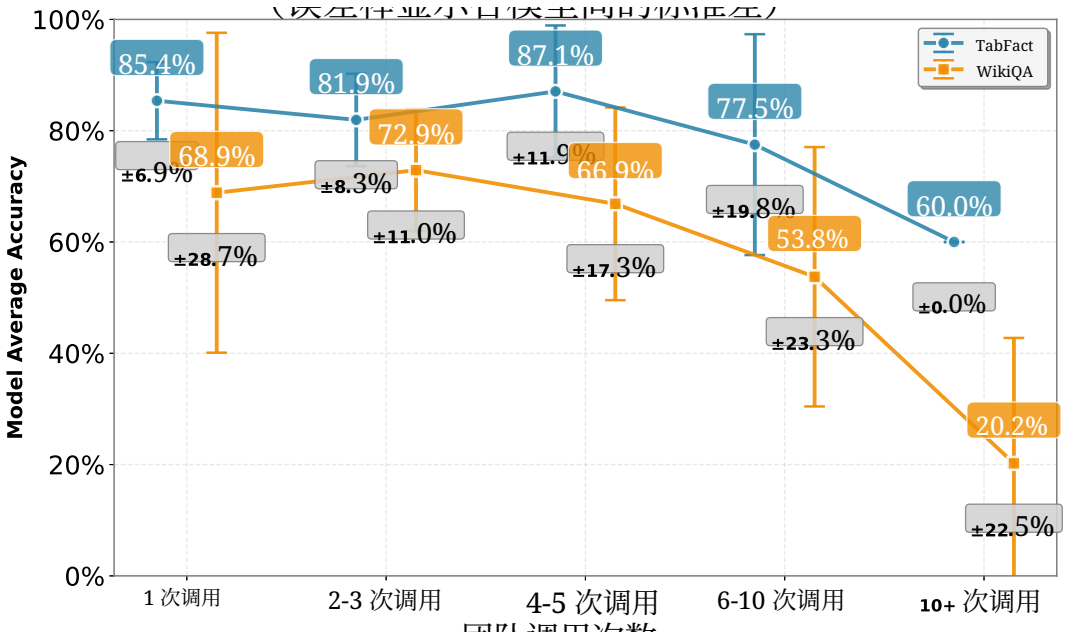
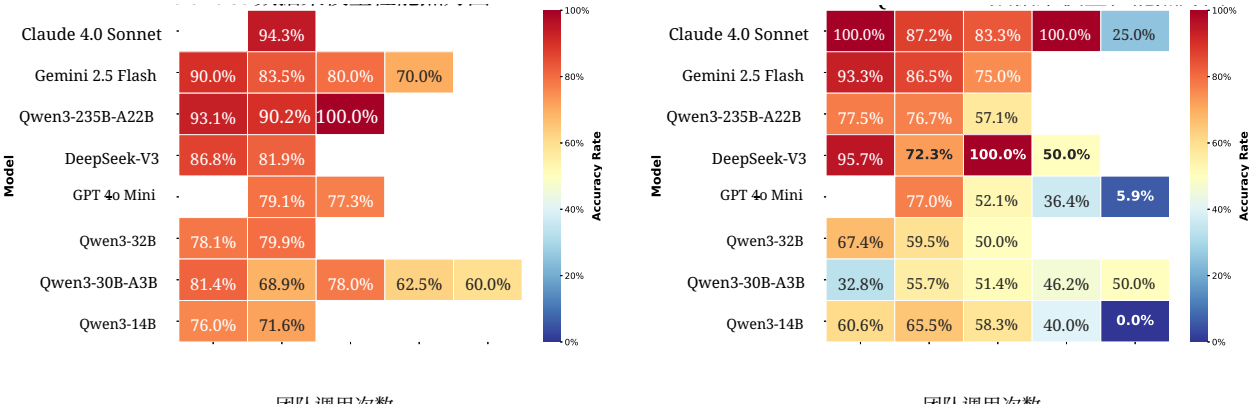


图 11：不同团队调用频率下模型的平均性能。误差棒表示所有模型的标准差。蓝线表示 TabFact 数据集的结果，橙线表示 WikiTQ 数据集的结果。



(a) TabFact 数据集　　(b) WikiTQ 数据集

图 12：展示不同团队协作调用频率下模型准确率的性能热力图。颜色深浅表示性能级别，红色越深表示准确率越高。每行代表一个不同的模型，每列代表一个不同的调用频率范围（1 次、2–3 次、4–5 次、6–10 次、 10+ 次）。

6. 理论与实践意义

我们提出的基于大语言模型的表格问答多智能体 DataFactory，为人工智能、数据库系统和自然语言处理的交叉领域提供了理论认知与实际应用方面的启示。我们的框架将传统的单智能体方法转变为专业化的团队协作范式，既推动了多智能体系统研究，也为表格数据分析应用带来了切实价值。

6.1. 理论贡献

从理论角度看，我们的工作拓展了对大语言模型多智能体协作的理解，展示了专业化团队架构如何克服单个大语言模型的根本局限。三方协作设计由采用 ReAct范式的数据负责人、数据库团队和知识图谱团队组成，为复杂推理场景中的任务分解与协调提供了启示。一项重要的理论贡献在于，我们提出了基于自然语言的协商机制，以灵活的智能体间通信取代僵化的工作流执行，从而能够在复杂推理过程中自适应地调整策略并共享知识。这一架构创新应对了平衡“专业化”与“协调”的理论挑战，表明通过智能编排和自然语义交互，可以系统地整合结构化数据处理（基于SQL）与关系知识表示（基于图）的互补优势。

自动化的“数据到知识图谱”转换算法为知识表示学习提供了一种方法。我们形式化了映射函数 $\mathcal{T} : \mathcal{D} \times \mathcal{S} \times \mathcal{R} \to \mathcal{G}$，并实现了实体构建和关系发现机制，为将结构化表格数据转换为语义知识图谱提供了一种系统化方法。这种方法不仅适用于表格问答，也为其他需要将结构化数据转换为关系表示以增强推理能力的领域提供了启示。

此外，我们针对 SQL 和 Cypher 生成提出的上下文增强提示策略，为如何整合外部知识、历史模式和领域信息系统以减少大型模型的幻觉提供了启示。通过将检索增强生成与专门的提示工程相结合，我们减少了幻觉并提高了查询生成的准确率。此外，实验结果揭示了协作频率与性能之间的倒U形关系，为多智能体系统中的有效智能体交互模式提供了经验证据。

6.2. 实践意义与平台集成

在实践层面，DataFactory 通过自然语言接口打破了企业数据分析中的关键壁垒，使非技术用户能够访问复杂的表格数据。该框架将数据库团队（负责结构化查询）与知识图谱团队（负责关系推理）结合起来，使用户无需具备专业的 SQL 或数据科学技能，即可同时探索数值模式和语义关联。

该平台实现了端到端集成工作流，涵盖数据摄取、知识图谱构建、查询处理和结果可视化。Web界面集成了多个功能模块，各专业智能体协同工作，自动完成 SQL 和 Cypher 查询、数据分析与可视化，从而大幅降低数据分析门槛并提升决策效率。

在实际部署中，业务分析师可以提出涉及跨表关系、时间模式和多跳推理的复杂查询。系统会返回结构化分析结果和知识图谱可视化，显著加快组织学习周期，并减少对专业数据团队的依赖。该平台采用模块化架构，确保在企业环境中具备可扩展性和自定义能力。平台支持双模式部署——基于开源模型的本地部署，以及基于商业 API 的云部署——以满足不同的资源和隐私要求。

除定量结果外，该平台还会生成解释性见解，提升自动化分析的透明度和可信度。这使用户不仅能够获得答案，还能了解数据库团队和知识图谱团队如何协同推理并得出结论。完整工作流程展示了结构化数据分析、知识关系挖掘和多智能体协同决策（界面演示见附录）。

7. 结论与未来工作

本文提出了一种基于大语言模型的多智能体 DataFactory 框架，通过系统化的任务分解和专业化团队协调，克服了单智能体表格问答方法的主要局限。该框架包含三项技术创新：（1）基于 ReAct 的数据负责人，用于协调数据库团队与知识图谱团队之间的智能任务分配；（2）自动化的数据到知识图谱转换算法，可捕捉结构关系和语义关系；（3）上下文增强工程策略，可提高 SQL 和 Cypher 操作的查询生成准确率。在三个基准数据集（TabFact、WikiTQ、FeTaQA）以及来自五家提供商的八种模型上进行的实验验证表明，该方法的性能持续提升：与基线方法相比，在 TabFact 上平均提升 20.2%，在 WikiTQ 上平均提升 23.9%；Cohen's d 值均大于 1，表明增强效果具有统计显著性。专业团队之间的协同协调优于单团队方法：在 TabFact 上提升 5.5%，在 WikiTQ 上提升 14.4%，在 FeTaQA 的 ROUGE-2 指标上最高提升 17.1%；其效果会根据不同模型的能力和任务复杂度相应变化。

未来研究应重点关注以下几个方面：面向大规模部署的计算效率优化；在流式工作负载和多租户工作负载下开展更加系统的可扩展性评估；引入自适应停止机制，以避免过度的多智能体交互；以及增强自动化知识图谱构建的稳健性。另一个有前景的方向是，将 DataFactory 作为后端推理服务，提供给更广泛的对话式智能体，包括面向任务的智能体和多模态对话系统。这些改进将在保留该框架处理复杂多跳推理任务的核心优势的同时，增强其实际应用能力。

A. 平台界面演示

DataFactory框架已实现为一个基于网络的平台，为数据探索和多智能体协作提供直观的界面。以下图示展示了三个主要组件的关键功能。

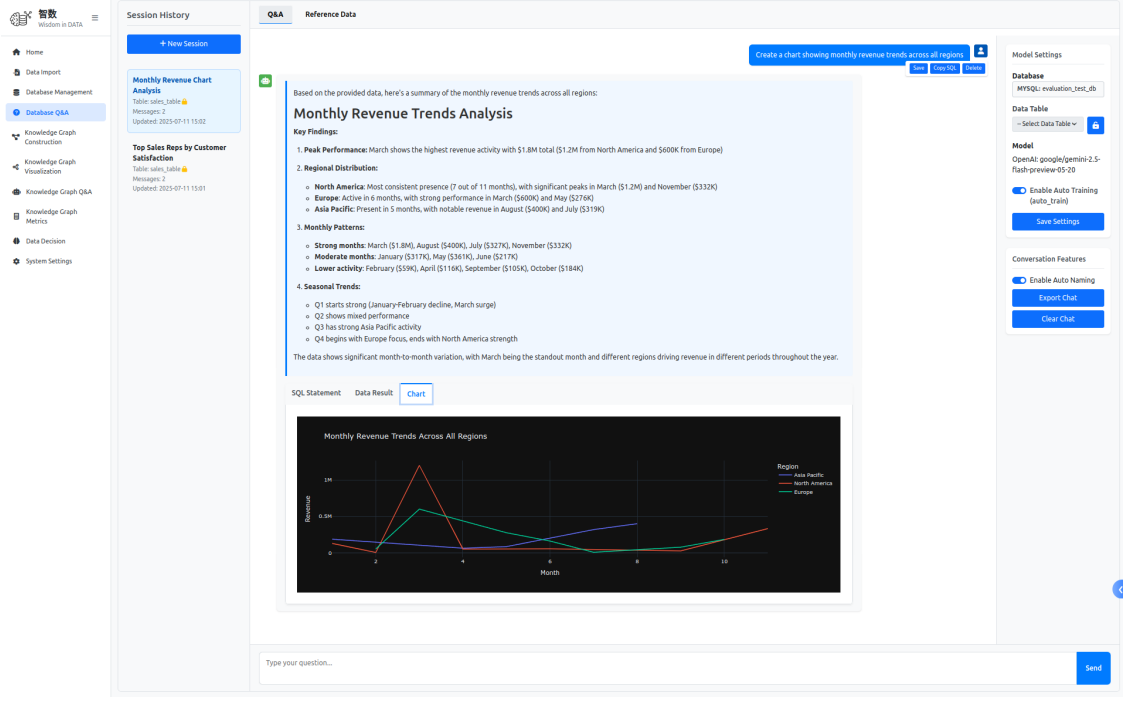


图 13： 数据库团队问答界面：用户通过自然语言提出跨地区月度收入趋势分析请求，系统自动生成 SQL 并将结果可视化。

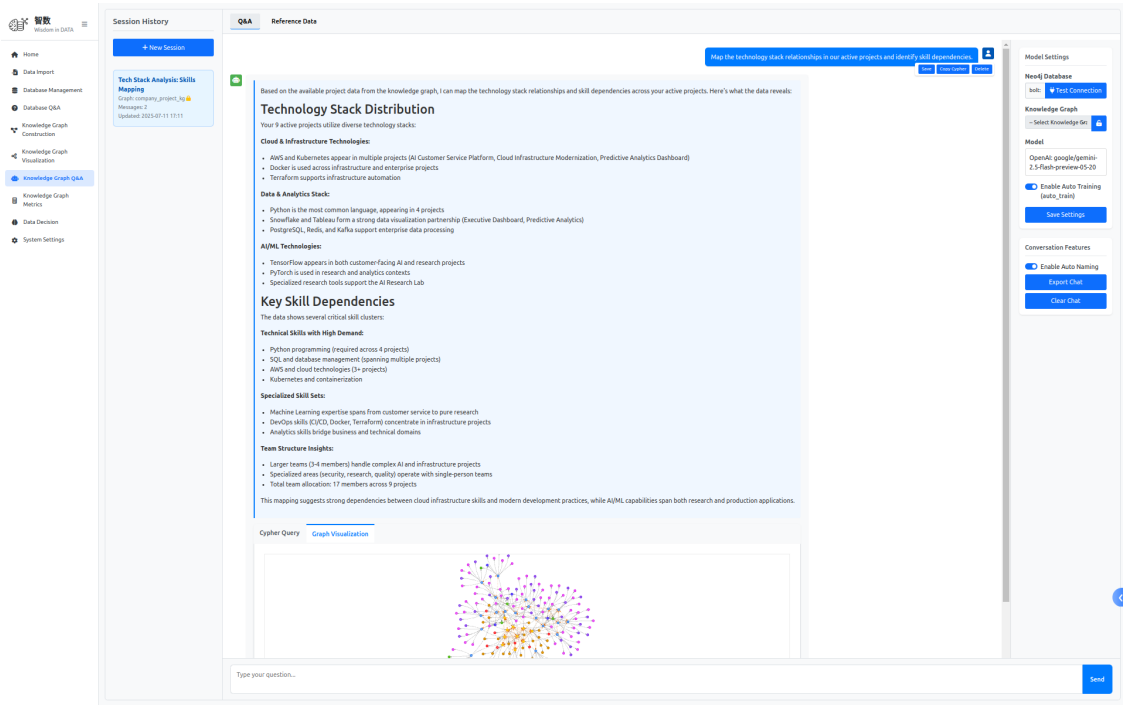


图 14： 知识图谱团队问答界面：用户提出技术栈关系和技能依赖分析请求，系统自动生成 Cypher 查询并将知识图谱子图可视化。

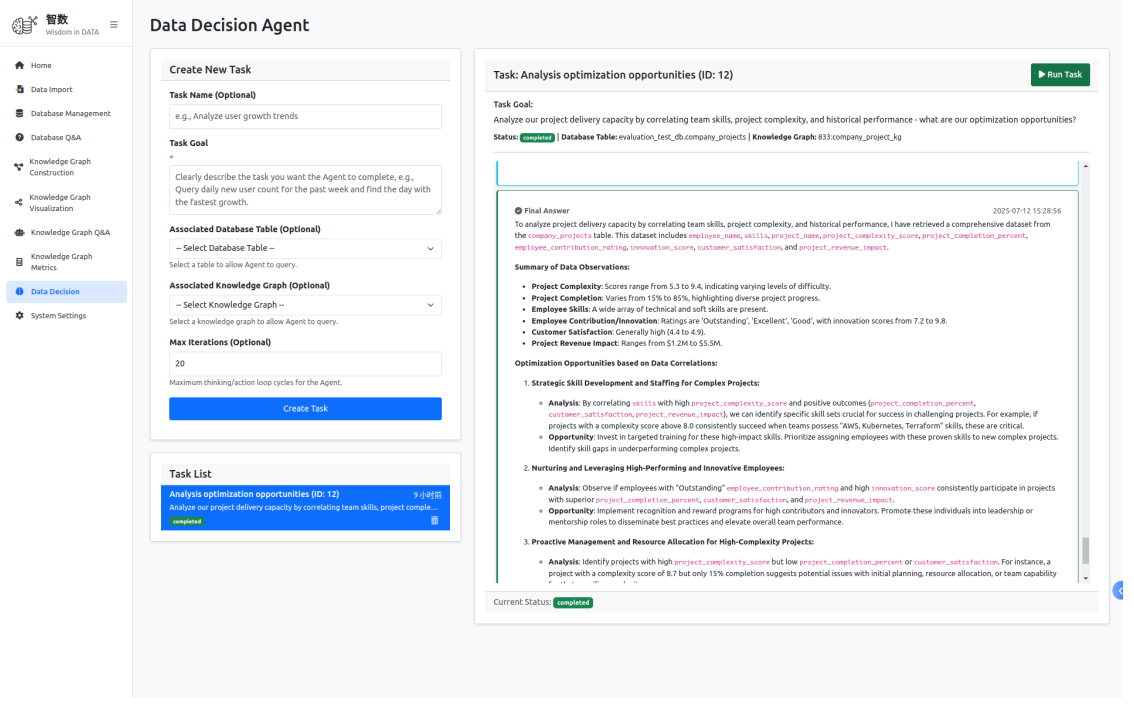


图 15： DataFactory 决策分析界面：在多智能体协作下，领导者协调数据库团队和知识图谱团队进行协同分析，并提供决策建议。

致谢

我们衷心感谢中国国防科技重点实验室基金（项目编号：6142006220506）为本研究提供的经费支持。我们还要感谢匿名审稿人提出的宝贵意见和建议，这些意见和建议显著提升了本文的质量。

CRediT 作者贡献声明

Tong Wang： 概念化、方法学、软件、调查、撰写初稿。**Chi Jin：** 数据整理、调查、撰写初稿、可视化。**Yongkang Chen：** 软件、验证、形式分析。**Huan Deng：** 软件、数据整理、资源。**XiaohuiKuang：** 项目管理、监督、撰写审阅与编辑。**Gang Zhao：** 概念化、项目管理、监督、撰写审阅与编辑。

参考文献

Anthropic，2024年。Introducing claude 4。https://www.anthropic.com/news/claude-4。Cao, Y., Chen, S., Liu, R., Wang, Z., Fried, D., 2023。面向多样表格结构问答的API辅助代码生成，载于：《2023年自然语言处理实证方法会议论文集》，第14536–14548页。https://doi.org/10.18653/v1/2023. emnlp-main.897。Chen, W., Chang, M.W., Schlinger, E., Wang, W.Y., Cohen, W.W., 2021。表格与文本上的开放域问答，载于：国际学习表征会议。https://openreview.net/forum?id=MmCRswl1UYl。Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., Wang, W.Y., 2020。Tabfact：用于基于表格的事实验证的大规模数据集，载于：国际学习表征会议。https://openreview.net/forum?id=rkeJRhNYDH。Cohen, J., 1988。《行为科学统计功效分析》。第2版。https://doi.org/10.4324/9780203771587。DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., 等，2025。Deepseek-v3技术报告。https://arxiv.org/abs/2412.19437，arXiv:2412.19437。Evangelatos, A., Filandrianos, G., Lymperaiou, M., Voulodimos, A., Stamou, G., 2025。Ails-ntua在SemEval-2025任务8中的表现：面向表格问答的语言到代码提示与错误修复。https://arxiv.org/abs/2503.00435，arXiv:2503.00435。Fan, M., Fan, J., Tang, N., Cao, L., Li, G., Du, X., 2025。AutoPrep：基于多智能体框架的自然语言问题感知数据准备。https://arxiv.org/abs/2412.10422，arXiv:2412.10422。Fang, X., Xu, W., Tan, F.A., Hu, Z., Zhang, J., Qi, Y., Sengamedu, S.H., Faloutsos, C., 2024。表格数据上的大语言模型（LLM）：预测、生成与理解——综述。《机器学习研究汇刊》https://openreview.net/forum?id= IZnrCGF9WI。Farquhar, S., Gal, Y., Rudner, T., 2024。使用语义熵检测大语言模型中的幻觉。Nature 630，625–630。https://doi.org/10.1038/s41586-024-07421-0。Google DeepMind，2024年。Gemini 2.5 flash：适用于日常任务的快速性能。https://deepmind.google/models/gemini/flash/。Guo, C., Tian, Z., Tang, J., Li, S., Wang, T., 2025。面向文本到SQL的多模式检索增强框架：结合庞加莱骨架检索与元指令推理。《信息处理与管理》62，103978。https://doi.org/10.1016/j.ipm.2024.103978。Herzig, J., Nowak, P.K., Müller, T., Piccinno, F., Eisenschlos, J., 2020。TaPas：通过预训练进行弱监督表格解析，载于：《计算语言学协会第58届年会论文集》，第4320–4333页。https://doi.org/10.18653/v1/2020.acl-main.398。Jiang, J., Zhou, K., Dong, Z., Ye, K., Zhao, X., Wen, J.R., 2023。StructGPT：用于大语言模型结构化数据推理的通用框架，载于：《2023年自然语言处理实证方法会议论文集》，第9237–9251页。https: //doi.org/10.18653/v1/2023.emnlp-main.574。Jin, N., Siebert, J., Li, D., Chen, Q., 2022。表格问答综述：最新进展，载于：《知识图谱与语义计算：知识图谱赋能数字经济》，第174–186页。https://doi.org/10.1007/978-981-19-7596-7_14。Katyayan, P., Joshi, N., 2022。基于规则的开放域问答系统在SQuAD v2.0数据集上的设计与开发。https: //arxiv.org/abs/2204.09659，arXiv:2204.09659。Khoja, R., Gupta, D., Fu, Y., Roth, D., Gupta, V., 2025。Weaver：交织SQL与大语言模型以实现表格推理。https://arxiv.org/abs/2505. 18961，arXiv:2505.18961。Lee, W., Kim, K., Lee, S., Lee, J., Kim, K.I., 2025。Piece of Table：用于在表格问答中选择子表的分治方法。https://arxiv.org/abs/2412.07629，arXiv:2412.07629。Li, Q., Huang, C., Li, S., Xiang, Y., Xiong, D., Lei, W., 2025。GraphOTTER：面向复杂表格问答的基于大语言模型的图推理演进，载于：《第31届国际计算语言学会议论文集》，第5486–5506页。https://aclanthology.org/ 2025.coling-main.368/。Liu, Q., Chen, B., Guo, J., Ziyadi, M., Lin, Z., Chen, W., Lou, J.G., 2022。TAPEX：通过学习神经SQL执行器进行表格预训练，载于：国际学习表征会议。https://openreview.net/forum?id=O50443AsCP。Lu, W., Zhang, J., Fan, J., Fu, Z., Chen, Y., Du, X., 2025。大语言模型表格处理：综述。《计算机科学前沿》19。https://doi.org/10.1007/s11704-024-40763-6。

Meier, S., Raut, P.N., Mahr, F., Thielen, N., Franke, J., Risch, F., 2025。基于结构化知识的因果发现：智能体思维流。《信息处理与管理》62，104202。https://doi.org/10.1016/j.ipm.2025.104202。Nan, L., Hsieh, C., Mao, Z., Lin, X.V., Verma, N., Zhang, R., Kryściński, W., Schoelkopf, H., Kong, R., Tang, X., Mutuma, M., Rosand, B., Trindade, I., Bandaru, R., Cunningham, J., Xiong, C., Radev, D., 2022。FeTaQA：自由形式表格问答。《计算语言学协会学报》10，35–49。https://doi.org/10.1162/tacl_a_00446。Neo4j，2012年。Neo4j——全球领先的图数据库。http://neo4j.org/。Nguyen, G., Brugere, I., Sharma, S., Kariyappa, S., Nguyen, A.T., Lecue, F., 2025。可解释的基于大语言模型的表格问答。《机器学习研究汇刊》https://openreview.net/forum?id=2eTsZBoU2W。Ojuri, S., Han, T.A., Chiong, R., Di Stefano, A., 2025。通过整合智能智能体与大语言模型优化文本到SQL转换技术。《信息处理与管理》62，104136。https://doi.org/10.1016/j.ipm.2025.104136。OpenAI，2024年。GPT-4o mini：推进高性价比智能。https://openai.com/index/ gpt-4o-mini-advancing-cost-efficient-intelligence/。Pal, V., Kanoulas, E., Yates, A., de Rijke, M., 2024。低资源印度语族语言的表格问答，载于：《2024年自然语言处理实证方法会议论文集》，第75–92页。https://doi.org/10.18653/v1/2024.emnlp-main.5。Pan, M.Z., Cemri, M., Agrawal, L.A., Yang, S., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Ramchandran, K., Klein, D., Gonzalez, J.E., Zaharia, M., Stoica, I., 2025。多智能体系统为何失败？，载于：ICLR 2025关于构建对语言模型及其应用的信任的研讨会。https://openreview.net/forum?id=wM521FqPvI。Pasupat, P., Liang, P., 2015。半结构化表格上的组合语义解析，载于：《计算语言学协会第53届年会及第7届自然语言处理国际联合会议论文集》（第1卷：长篇论文），第1470–1480页。https://doi.org/10.3115/v1/P15-1142。Qin, B., Hui, B., Wang, L., Yang, M., Li, J., Li, B., Geng, R., Cao, R., Sun, J., Si, L., Huang, F., Li, Y., 2022。文本到SQL解析综述：概念、方法与未来方向。https://arxiv.org/abs/2208.13629, arXiv:2208.13629。Sairaj, R.T., Balasundaram, S., 2025a。面向检索增强建模的本体映射：减少基于预训练语言模型自动生成问题中的事实性幻觉。《应用本体》20，69–88。https://doi.org/10.1177/15705838251343009。Sairaj, R.T., Balasundaram, S.R., 2025b。使用RAG模型的集成学习减少自动生成考试问题中的重复问题主题。《发现计算》28，161。https://doi.org/10.1007/s10791-025-09683-2。Sui, Y., Zhou, M., Zhou, M., Han, S., Zhang, D., 2024。表格遇上大语言模型：大语言模型能理解结构化表格数据吗？一项基准测试与实证研究，载于：《第17届ACM Web搜索与数据挖掘国际会议论文集》，第645–654页。https://doi.org/10.1145/3616855.3635752。Wang, Y., Gan, J., Qi, J., 2025a。Tabsd：基于SQL的表格分解的大型自由形式表格问答。https://arxiv.org/ abs/2502.13422，arXiv:2502.13422。Wang, Z., Zhang, R., Nie, Z., 2025b。通过答案公式联合生成实现通用表格问答。https://arxiv.org/abs/2503. 12345，arXiv:2503.12345。Wei, X., Croft, B., McCallum, A., 2006。用于答案检索的表格抽取。《信息检索》9，589–611。https://doi.org/10.1007/ s10791-006-9005-5。Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., 等，2025a。Qwen3技术报告。https://arxiv.org/abs/2505.09388，arXiv:2505.09388。Yang, Z., Chen, L., Cohan, A., Zhao, Y., 2025b。Table-r1：表格推理的推理时扩展。https://arxiv.org/abs/2505.23621。Yang, Z., Du, Z., Zhang, M., Du, W., Chen, J., Duan, Z., Zhao, S., 2025c。三元组是关键：结构化使基于大语言模型的表格问答中的分解与验证更容易，载于：第十三届国际学习表征会议。https://openreview.net/forum? id=UwcZEoNP19。Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y., 2023。ReAct：在语言模型中协同实现推理与行动，载于：第十一届国际学习表征会议。https://openreview.net/forum?id=WE_vluYUL-X。Yin, P., Neubig, G., Yih, W.t., Riedel, S., 2020。TaBERT：文本与表格数据联合理解的预训练，载于：《计算语言学协会第58届年会论文集》，第8413–8426页。https://doi.org/10.18653/v1/2020.acl-main.745。Yu, P., Chen, G., Wang, J., 2025。Table-critic：用于表格推理中协作式批评与优化的多智能体框架。https: //arxiv.org/abs/2502.11799，arXiv:2502.11799。Zhang, T., Yue, X., Li, Y., Sun, H., 2024a。TableLlama：迈向面向表格的开放式大型通用模型，载于：《2024年北美计算语言学协会分会会议：人类语言技术论文集》（第1卷：长篇论文），第6024–6044页。https://doi.org/10.18653/v1/2024.naacl-long.335。Zhang, Y., Henkel, J., Floratou, A., Cahoon, J., Deep, S., Patel, J.M., 2024b。ReAcTable：增强ReAct以实现表格问答。《VLDB学报》17，1981–1994。https://doi.org/10.14778/3659437.3659452。Zhang, Z., Gao, Y., Lou, J.G., 2024c。 e^5：通过解释、抽取、执行、展示与外推，利用增强型大语言模型进行零样本层次化表格分析，载于：《2024年北美计算语言学协会分会会议：人类语言技术论文集》（第1卷：长篇论文），第1244–1258页。https://doi.org/10.18653/v1/2024.naacl-long.68。Zhao, B., Ji, C., Zhang, Y., He, W., Wang, Y., Wang, Q., Feng, R., Zhang, X., 2023。大语言模型是复杂表格解析器，载于：《2023年自然语言处理实证方法会议论文集》，第14786–14802页。https://doi.org/10.18653/v1/2023. emnlp-main.914。Zhao, J., Zhang, P., Wang, Y., Xin, R., Lu, X., Li, R., Lyu, S., Ou, Z., Song, M., 2025。TqAgent：结合知识图谱与树结构推理增强基于表格的问答。《应用科学》15。https://doi.org/10.3390/app15073788。Zhou, W., Mesgar, M., Friedrich, A., Adel, H., 2025。通过工具使用实现复杂表格问答中的高效多智能体协作与在线规划，载于：《计算语言学协会会议论文集：NAACL 2025》，第945–968页。https://doi.org/10.18653/v1/

2025年.findings-naacl.54. Zhu, J.P., Cai, P., Xu, K., Li, L., Sun, Y., Zhou, S., Su, H., Tang, L., Liu, Q., 2024年。Autotqa：通过多智能体大语言模型实现自主表格问答。《VLDB学报》17，3920–3933。 https://doi.org/10.14778/3685800.3685816. Ziqi, J., Lu, W., 2023年。Tab-CoT：零样本表格思维链，载于：计算语言学协会会议论文集：ACL 2023，页码 10259–10277。 https://doi.org/10.18653/v1/2023.findings-acl.651。