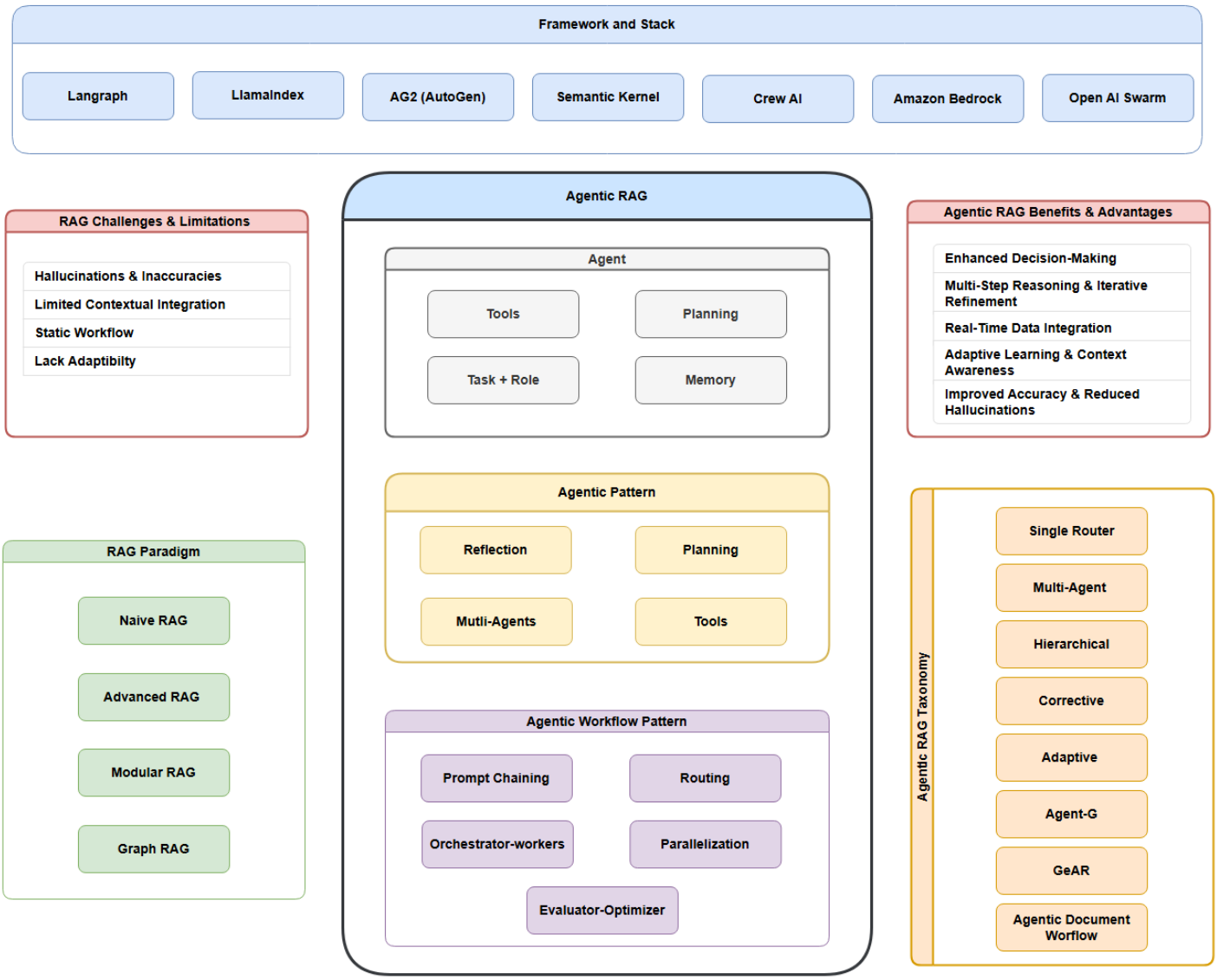
Agentic RAG

论文：《AGENTIC RETRIEVAL-AUGMENTED GENERATION: A SURVEY ON AGENTIC RAG》

Agentic RAG 将AI Agent融入了 RAG，采用了Agent中的思想，例如Reflection、planning、工具使用、多Agent协作等，在以下方面都表现出卓越的性能：

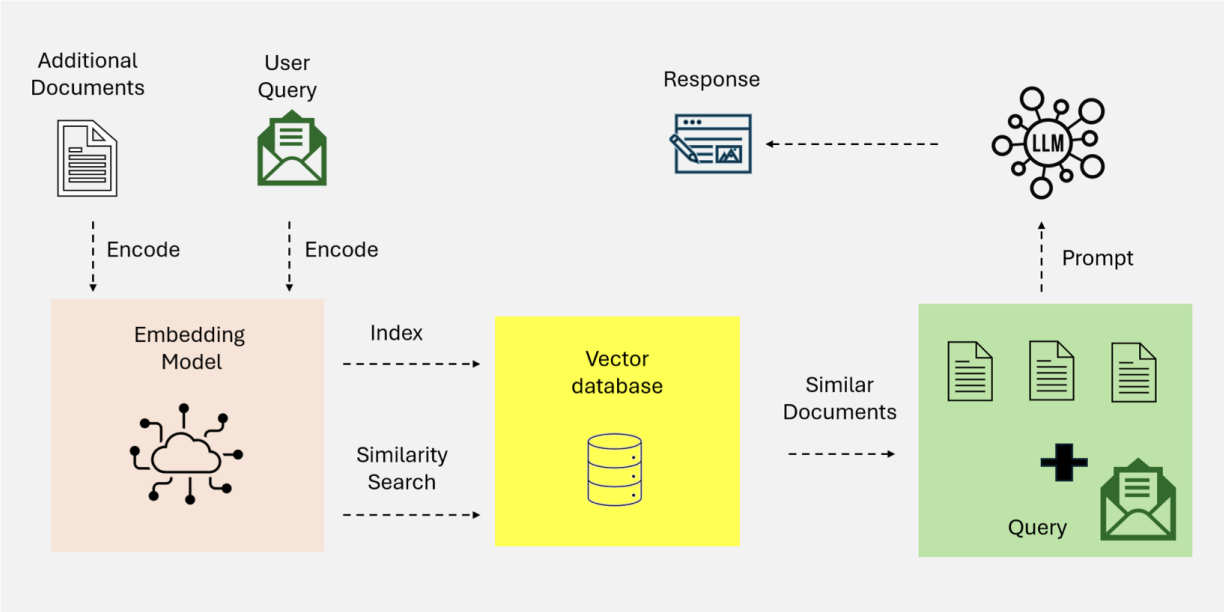
- 多领域知识检索。
- 以文档为中心的实时工作流程。
- 可扩展、自适应且合乎道德的 AI 系统。



RAG 发展历程

Naive RAG

Naive RAG 是最基础的一种架构，用于结合检索和生成来处理复杂的任务。下图中的示例依赖于基于关键词的检索技术，如 **TF-IDF 和 BM25**，从静态数据集中获取文档，用于增强模型的生成能力。（Naive RAG也支持向量检索）

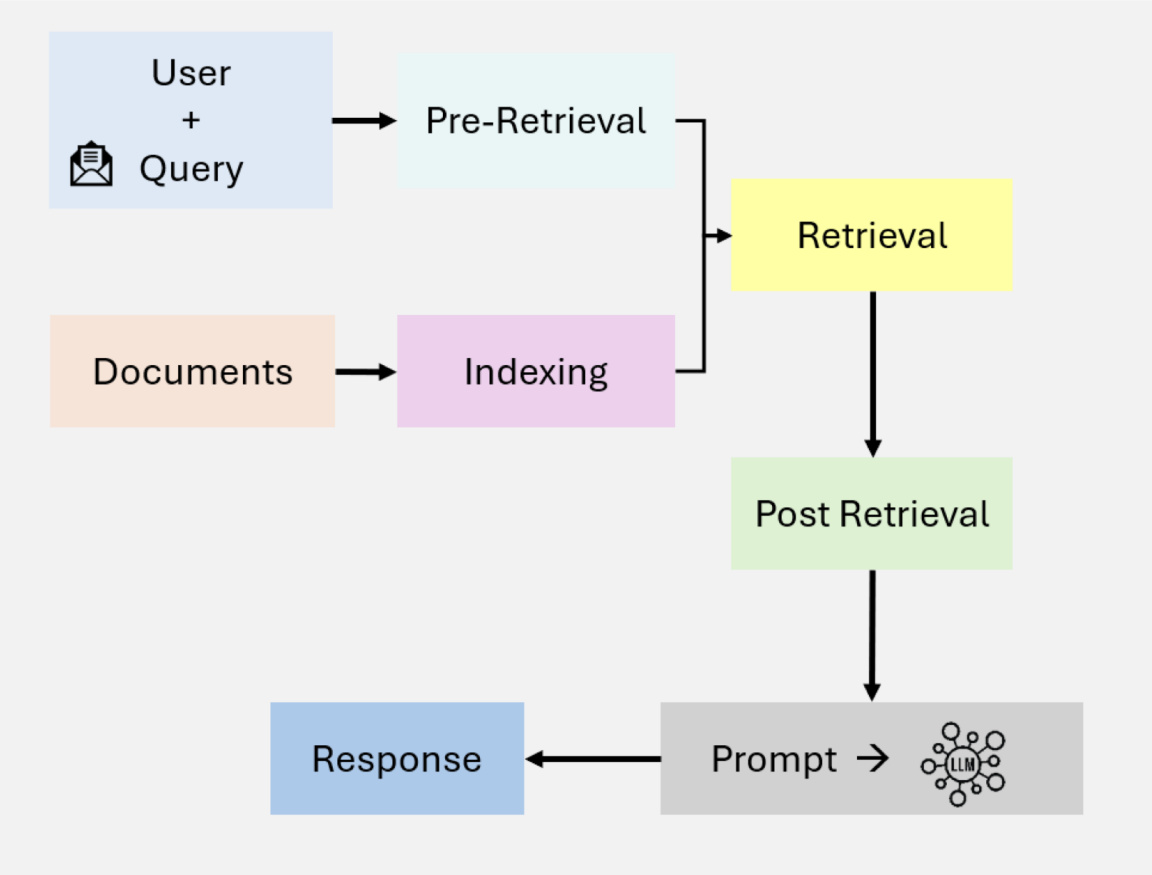


Naive RAG容易实现，适用于简单的事实查询或上下文复杂性低的任务，但存在以下缺陷：

- **缺乏上下文意识**：由于依赖词汇匹配而非语义理解，检索到的文档往往无法捕捉查询的语义细微差别
- **输出碎片化**：缺乏数据预处理或上下文整合，往往导致回答不连贯或过于通用
- **可扩展性问题**：基于关键词的检索技术在处理大型数据集时存在困难，往往无法识别最相关的信息

Advanced RAG

Advanced RAG 融入语义理解和增强的检索技术。使用**密集检索模型（如 Dense Passage Retrieval，DPR）**和**神经排序算法**来提高检索精度。



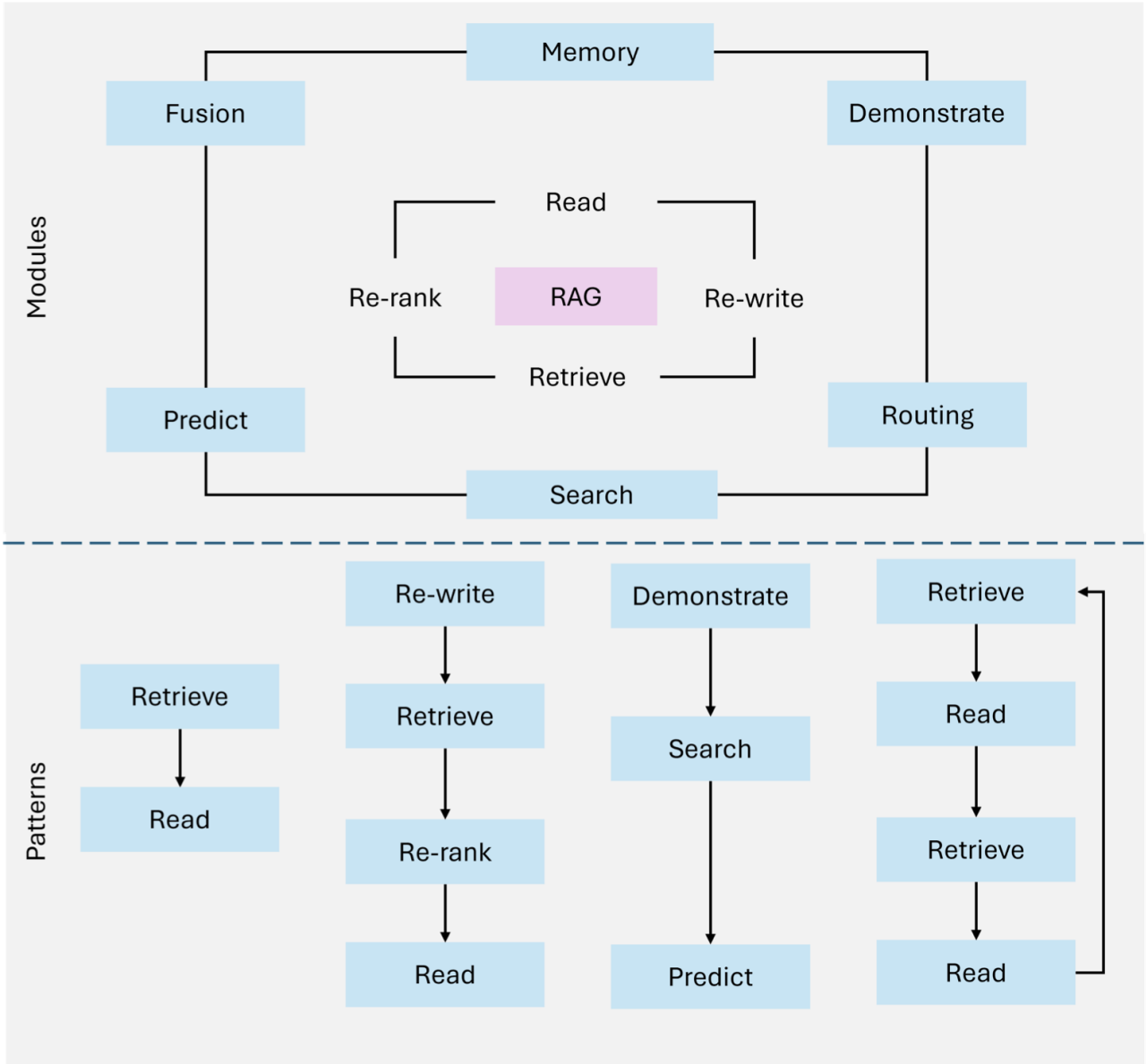
Advanced RAG 的核心特性包括：

- **密集向量搜索**：查询和文档以高维向量空间表示，从而实现用户查询和检索到的文档之间更好的语义对齐
- **上下文重新排序**：神经模型重新对检索到的文档进行排序，以优先考虑最相关的上下文信息
- **迭代检索**：Advanced RAG 引入了多跳检索机制，使得在复杂查询中可以跨多个文档进行推理

Advanced RAG 适用于需要高精度和细致理解的应用，例如研究综述和个性化推荐。然而，仍然存在一些挑战，比如计算开销和有限的可扩展性，特别是在处理大型数据集或多步查询时。

Modular RAG

Modular RAG 强调**灵活性**和**定制性，像“搭积木”一样组装AI能力**。将检索和生成流程分解为独立、可重用的组件，从而实现领域特定的优化和任务适应性。下图展示了 Modular RAG 的架构，展示了混合检索策略、可组合的流程和外部工具集成。



Modular RAG 的关键创新包括：

- **混合检索策略**：**将稀疏检索方法**（例如 稀疏编码器-BM25）**与密集检索技术**（例如 DPR - Dense Passage Retrieval）相结合，准确性更高（就是bge里的混合检索）。
- **工具集成**：将外部 API、数据库、计算工具等功能纳入系统，用于处理特定任务，如实时数据分析或领域特定计算
- **可组合的流程**：Modular RAG 使得检索器、生成器和其他组件可以独立替换、增强或重新配置，从而实现对特定用例的高度适应性

二、具体案例1：电商平台“个性化商品推荐”系统

业务需求：用户搜索“适合跑步的透气运动鞋”，需结合商品描述、用户历史购买记录、实时促销活动推荐商品。

Modular RAG架构拆解：

1. **检索模块（可替换）：**
   - **稀疏检索组件**：用BM25算法匹配“跑步”“透气”“运动鞋”关键词，从商品数据库快速捞出100个候选商品（适合精确匹配）。
   - **密集检索组件**：用DPR模型将用户查询转为向量，在商品描述向量库中找语义相似的商品（比如用户实际需要“马拉松跑鞋”，但没明确说“马拉松”，靠语义匹配补充）。
   - **混合策略**：先跑BM25过滤出基础候选，再用DPR做语义精排，效率比单一检索提升30%。

2. **工具集成模块：**
   - **实时数据工具**：调用促销API，筛选出“当前打折”的候选商品。
   - **用户画像工具**：从CRM系统获取用户历史购买记录（比如用户曾买过“某品牌跑步袜”），通过协同过滤算法提升推荐精准度。

3. **生成模块（领域定制）：**
   - 输出结构化推荐列表，包含：

```
代码块
1  推荐商品：XX品牌透气马拉松跑鞋（折扣价¥899，比用户历史购买均价低20%）
2  推荐理由：含GORE-TEX透气技术（来自商品描述检索），用户同品牌复购率达45%（来自用户画像工具）
3  购买链接：XXX（调用电商API生成）
```

传统RAG vs Modular RAG对比：

- 传统RAG：只能按“关键词检索→生成推荐语”固定流程，无法动态加入“促销筛选”和“用户画像”，推荐结果千篇一律。
- Modular RAG：通过替换检索模块（增加DPR语义匹配）、插入促销API和用户画像工具，推荐点击率提升40%，且能快速复用至“运动服装”“健身器材”等新类目。

通过 **标准化接口+插件化设计**，让模块像“USB设备”一样即插即用：

4. **输入输出标准化：**

- 检索模块输出统一格式（如JSON：{实体:光伏补贴，内容:补贴延长3年，来源:政策文件第X条}）
- 工具模块接收固定参数（如财务模型工具统一接收“股价”“营收增长率”等字段）

5. **流程编排可视化**：使用低代码平台（如LlamaIndex的Agentic Workflows），通过拖放组件定义流程：

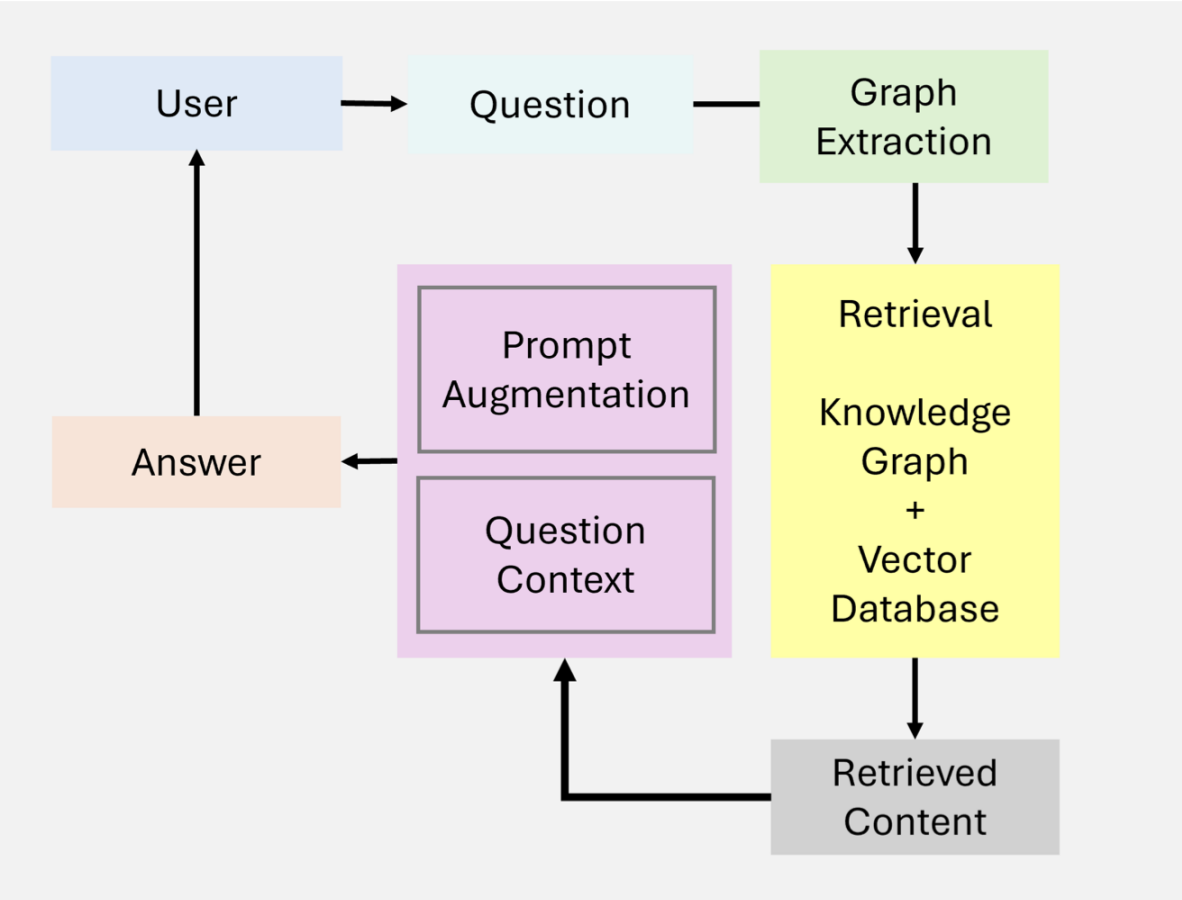
```
用户查询 → [关键词检索] → [语义精排] → [财务模型计算] → [生成报告]
```

6. **模块热插拔**：当发现“某电商的推荐算法效果差”，可随时替换为“协同过滤算法模块”，无需中断服务，就像给电脑换显卡一样方便。

本质上，Modular RAG解决了传统RAG的“一刀切”问题，让AI系统从“定制化手工艺品”变成“标准化组装件”，尤其适合企业级复杂场景——就像乐高积木，用有限的模块组合出无限的可能。

Graph RAG

Graph RAG整合基于图的数据结构扩展了传统的检索增强生成系统，利用图数据中的关系和层次结构，增强了多跳推理和上下文丰富性。通过引入基于图的检索，Graph RAG能够实现更丰富、更准确的生成输出，特别是在需要关系理解的任务中。



Graph RAG的特点包括：

- **节点连接性**：捕获并推理实体之间的关系。
- **层次知识管理**：通过基于图的层次结构处理结构化和非结构化数据。
- **上下文丰富**：通过利用基于图的路径添加关系理解。

然而，Graph RAG也有一些局限性：

- **可扩展性有限**：依赖图结构可能会限制可扩展性，特别是在数据源广泛时。
- **数据依赖性**：高质量的图数据对于有意义的输出至关重要，限制了其在非结构化或注释不佳的数据集中的适用性。
- **集成复杂性**：将图数据与非结构化检索系统集成增加了设计和实现的复杂性。

Graph RAG适用于医疗诊断、法律研究等需要对结构化关系进行推理的应用领域。更详细描述见3.6 GraphRAG。

> 📚 **传统RAG面临的问题**
>
> - **上下文整合**：通常难以将RAG检索到的内容其无缝地融入生成的响应中，检索是静态的，缺乏上下文意识，导致输出结果零散、不一致或者过于通用。
> - **多步推理**：复杂的查询需要多步查询和推理，传统的 RAG 系统往往无法根据中间洞察或用户反馈来优化检索，导致响应不完整或不连贯。

- **可拓展性和高延迟**：随着数据源的增多，查询将需要更大的计算资源，导致了显著的延迟。

Agentic RAG

Agentic RAG 引入了具有动态决策能力和工作流优化能力的自主代理，实现了范式转变。与静态系统不同，Agentic RAG 采用**迭代改进**和**自适应检索策略**来应对复杂、实时和多领域查询。这种范式利用了检索和生成过程的模块化，同时引入了基于代理的自治性。

Agentic RAG 的关键特性包括：

- **自主决策**：代理根据查询的复杂性独立评估和管理检索策略
- **迭代改进**：引入反馈循环以提高检索准确性和响应相关性
- **工作流优化**：动态编排任务，实现实时应用的高效性

Agentic RAG 面临的挑战：

- **协调复杂性**：管理代理之间的交互需要复杂的编排机制
- **计算开销**：使用多个代理增加了复杂工作流的资源需求
- **可扩展性限制**：虽然具有可扩展性，但系统的动态性可能会对高查询量的计算资源造成压力

Agentic RAG 在客户支持、金融分析和自适应学习平台等领域表现出色，其中动态适应性和上下文精确性至关重要。

类型	检索方式	核心创新	适用场景
Naive RAG	关键词匹配	初步结合检索与生成	简单事实问答
Advanced RAG	密集向量+神经排序	语义对齐与多跳推理	研究分析、推荐系
Modular RAG	混合检索+模块化	灵活定制与工具集成	跨领域复杂任务
Graph RAG	图结构推理	关系与层次化知识管理	医疗诊断、法律研
Agentic RAG	动态Agent协作	自主决策与迭代优化	实时交互场景（如

Agentic RAG 将AI Agent融入了 RAG，通过引入Agent的自主规划、动态决策和反思能力，显著提升了复杂任务的解决能力。

本文梳理了Naive RAG -> Advanced RAG -> Modular RAG -> Graph RAG -> Agentic RAG的发展路径。

举个例子对比Agentic RAG与传统RAG的区别：

传统RAG在电商客服的应用

1. **数据输入**：电商公司将产品、订单、售后等文档录入普通RAG系统建知识库。
2. **客户咨询**：客户问“衣服尺码不合适，咋退换货？”

3. **回复客户**：系统检索知识库，给出“在订单详情页点退换货申请按钮，提交信息后处理”的回复。若客户追问复杂问题，如“提交后提示不符条件，为啥？”，给出的回复可能模版痕迹重、无法真正解决问题。

Agentic RAG在电商客服的应用

4. **准备数据与规则**：录入文档，并为Agentic RAG系统设定Agent的执行规则，如依反馈优化、多步骤推理策略。
5. **客户首次咨询**：客户问“衣服尺码不合适，咋退换货？”，系统检索后回复“在订单详情页点退换货申请按钮，提交信息后处理”。
6. **客户追问**：客户追问“提交后提示不符条件，为啥？”，Agent启动多步骤推理，查看订单信息，对比政策，回复“超退换货期限。若有质量问题，提供证明可处理”。
7. **学习优化**：如果客户满意，系统就强化策略；如果客户继续问“质量问题咋证明？”，系统依反馈优化回复，下次提供更详细内容。
8. **多问题处理**：客户同时问“商品啥时发货？有类似推荐吗？”，不同Agent分工，分别查发货时间与推荐商品，整合回复“预计XX发货，推荐XX、XX商品”。