

AtomicRAG: 原子—实体图用于检索增强生成

Yanning Hou^{1*} Duanyang Yuan^{1*} Sihang Zhou^{1†} Xiaoshu Chen¹ Ke Liang¹
Siwei Wang¹ Xinwang Liu¹ Jian Huang¹

Abstract

最近的 GraphRAG 方法将图结构融入文本索引与检索中，利用知识图谱三元组连接文本块，从而提升检索的覆盖范围和准确率。然而，我们观察到，将文本块视为知识表征的基本单元会僵化地将多个原子事实捆绑在一起，限制了支持多样化检索场景所需的灵活性与适应性。此外，基于三元组的实体链接对关系抽取错误敏感，可能导致推理路径缺失或错误，最终损害检索准确率。为解决这些问题，我们提出原子-实体图（Atom-Entity Graph），一种更精确、更可靠的知識表征与索引架构。在我们的方法中，知识以知识原子的形式存储，即独立且自包含的事实单元，而非粗粒度的文本块。这使得知识元素能够灵活重组而不相互干扰，从而实现与多样查询视角的无缝对齐。实体间的边仅表示关系是否存在。通过结合个性化 PageRank 与基于相关性的过滤机制，我们保持了准确的实体连接，并提升了推理的可靠性。理论分析及在五个公开基准上的实验表明，所提出的 AtomicRAG 算法在检索准确率和推理鲁棒性方面均优于强基线 RAG 方法。代码：<https://github.com/7HHHHH/AtomicRAG>.

*Equal contribution ¹National University of Defense Technology, China. Correspondence to: Sihang Zhou <sihangjoe@gmail.com>.

1. 引言

检索增强生成 (RAG) (Lewis et al., 2020; Gao et al., 2023) 已成为连接大模型 (LLMs) (Guo et al., 2025a; Yang et al., 2024) 与外部语料库以完成知识密集型任务的标准范式，提升了事实依据性和答案准确率。经典的 RAG 流水线 (Karpukhin et al., 2020; Izacard & Grave, 2020) 依赖于基于块的检索：将文档分割为固定长度的文本块，进行嵌入，并使用稠密相似度搜索进行检索。这种简单而高效的设计在很大程度上保留了原始语义，但将知识视为孤立的片段，忽略了块之间的内部关系，且常引入冗余上下文，导致在需要整合分散证据或遵循多步推理链的查询上表现脆弱。

另一种方法分支是图结构 RAG 算法，这类算法通常采用两种知识组织方式之一：要么完全用基于三元组的图来替代原始语料库，作为主要的知识存储库 (Zhang et al., 2025; Hu et al., 2024; Wang et al., 2025; Mavromatis & Karypis, 2025; Peng et al., 2024; Chen et al., 2025)，要么将知识图谱与语料库中的文本块进行关联。然而，三元组替换策略在简化过程中不可避免地丢弃了大量上下文信息，而这些信息对于准确的问答系统往往至关重要。与此同时，图-块关联策略与传统的基于块的检索方法类似，将知识限制在固定的段落中，限制了其根据不同的查询需求动态重组信息的能力，可能妨碍对相关内容的精确检索。此外，在开放领域设置下提取可靠的三元组关系仍然是一项挑战。三元组构建中的错误可能导致检索过程中推理路径不完整或不正确，最终影响生成答案的质量。如图 1 所示，GraphRAG 所采用的图结构虽然看起来索引良好，但作为知识表征却不可靠。例如，将“基底细胞皮肤癌”的宿

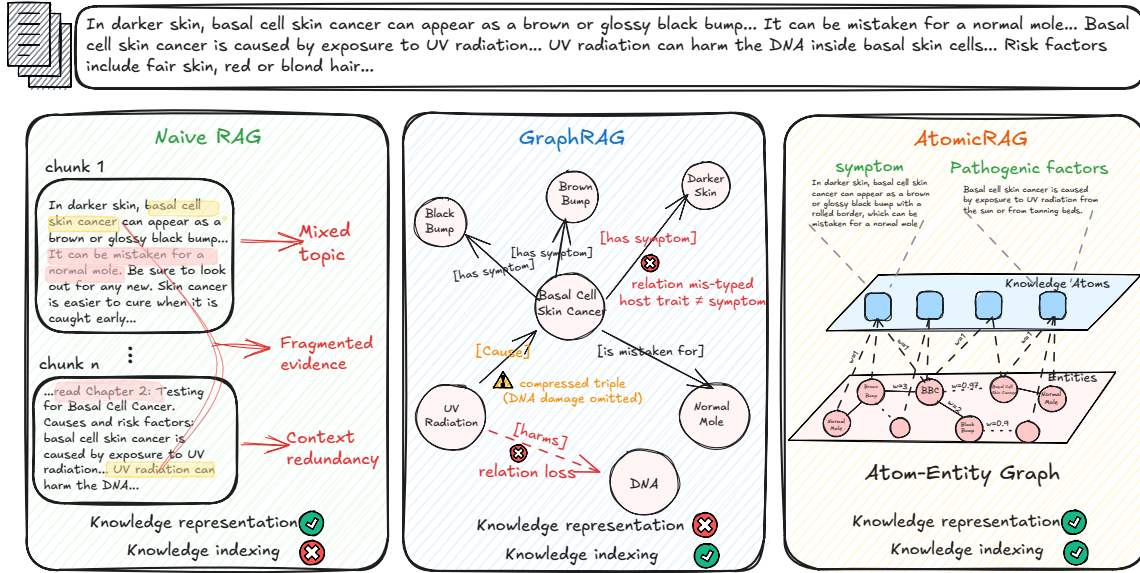


Figure 1. 三种方法的知识表征与索引比较。原生 RAG 使用粗粒度文本块作为基本存储单元，并通过语义相似度进行索引。GraphRAG 通过三元组或块级结点组织知识，利用关系边建立连接，以实现全局索引。所提出的原子-实体图则以细粒度的知识原子表示语料库，通过共现关系连接实体，从而在知识片段之间产生更稳定、更准确的连接。

主属性错误地归类为疾病症状（如将基底细胞皮肤癌与较深肤色关联），将涉及紫外线暴露和 DNA 损伤的多步因果解释压缩成单一粗粒度关系，甚至完全缺失关键的机制性连接。因此，由这种索引引导的检索遵循的是结构上看似合理但信息严重失真的路径。

为解决现有知识组织与索引方法的局限性，本文提出了一种名为 AtomicRAG 的新颖检索增强框架，其核心是一个原子-实体图（AEG）。在预处理阶段，语料库被分解为细粒度、自包含的单元，称为知识原子，作为信息的基本表示。AEG 通过未标记的边结构化地组织这些原子及其提取出的实体，利用共现关系捕捉实体间（相关性边）以及原子与其所含实体间（包含性边）的关系。这种基于图的表示方式支持灵活且精确的检索，无论是在局部还是全局范围内，同时为语义搜索提供了稳定可靠的结构。

在检索阶段，AtomicRAG 采用查询分解策略，将推理过程与检索解耦。复杂查询被自适应地拆分为与原子对齐的子问题，从而实现与知识库的细粒度匹配。随后，一种实体共振图检索机制结合语义相似度与基于图的相关性传播，识别出最相关的原子。

最后，通过过滤步骤去除冗余或无关内容，确保仅将所有子问题收集到的简洁、高价值证据传递给语言模型。该情景不仅降低了检索过程中的噪声，还提升了生成答案的事实准确率与来源透明性。

本文的贡献有三方面：(1) 我们提出了原子-实体图（AEG），这是一种新型的知识表征方法，相较于传统的基于块或关系标注的图结构，具有更高的灵活性和鲁棒性。(2) 我们设计了一种查询自适应的检索流水线，首先将复杂问题分解为原子子问题，然后利用实体共振图传播机制，精准地收集简洁且相关的证据。(3) 通过理论分析及在五个基准上的大量实验，我们证明了 AtomicRAG 在检索准确率和推理鲁棒性方面均优于强基准模型，尤其在需要证据组合的多跳查询中表现更优。

2. 相关工作

2.1. 检索增强生成

检索增强生成(RAG)通过检索外部证据并基于检索到的上下文来条件化生成，使大语言模型 (LLM) 的输出具有依据 (Qian et al., 2024; Hou et al., 2025)。除了段落级别的索引之外，稠密检索与外部知识结

合 (Dense X Retrieval) 表明, 在固定计算预算下, 命题级别的检索能够提升检索质量以及下游问答性能 (Chen et al., 2023)。为减少当检索到的段落脱离其原始文档上下文时产生的分歧, 上下文检索 (Contextual Retrieval) 为每个文本块自动添加特定于该块的上下文信息, 适用于稠密检索和 BM25 检索, 并进一步受益于重排序机制。针对多步骤信息需求, HyDE 通过假设性文档嵌入丰富查询 (Gao et al., 2022); 而 IRCot, Iter-RetGen 以及诸如 MDR 等多跳稠密检索器, 则将检索与中间推理信号交错或迭代进行, 逐步定位支持性证据 (Trivedi et al., 2022; Shao et al., 2023; Xiong et al., 2020)。其他补充性工作如 RAPTOR 和 LLMingua 通过层次化组织和提示压缩提升了长上下文的可用性, 而 Self-RAG 则在解码过程中通过自我批判实现按需检索 (Sarathi et al., 2024; Jiang et al., 2023; Asai et al., 2023)。尽管取得了这些进展, 但在多跳情景下, 构建稳定且跨文档的证据链仍具挑战性。

2.2. 基于图形的 RAG

基于图形的 RAG (Luo et al., 2025a; Guo et al., 2025b; Luo et al., 2025b) 通过图结构组织证据单元和实体关联, 将检索从基于相似度的 top-k 块扩展为可组合的子图或路径级证据。微软的 GraphRAG (Edge et al., 2024) 利用语言模型构建图结构, 并借助社区级摘要强化跨文档聚合与查询聚焦的综合能力。HippoRAG (Gutierrez et al., 2024; Gutierrez et al., 2025) 将知识图谱与个性化 PageRank 相结合, 从查询种子在图上进行传播, 实现单次检索过程中的多跳信息整合。LightRAG (Guo et al., 2024) 引入图结构与两阶段检索流水线, 在覆盖范围与效率之间取得平衡。GFM-RAG (Luo et al., 2025c) 进一步采用图神经网络, 增强对结构化知识的多跳推理能力。总体而言, 基于图形的方法的优势往往取决于高质量图结构的构建: 实体覆盖、边的正确性与一致性, 以及持续更新的成本直接影响图遍历与路径组合的可靠性; 当图存在噪声或不完整时, 多跳传播与拼接可能放大错误, 导致证据链不稳定。

3. 方法

图 2 展示了 AtomicRAG 的整体架构, 其运行分为四个阶段。(i) 原子-实体图构建 (离线), 用于构建持久化的原子-实体图 (AEG) 作为知识存储; (ii) 原子级问题分解, 将 q 重写为原子级别的子问题; (iii) 实体共振图检索, 通过 AEG 传播信号以选择候选原子; 以及 (iv) 原子筛, 对原子进行过滤和排序, 以形成紧凑的上下文供下游生成使用。这四个阶段共同构成一个端到端的流水线, 用于组合多跳证据以回答复杂查询。

3.1. 原子-实体图构建

知识原子与三元组的联合抽取。 给定语料库 $\mathcal{C} = \{d_i\}_{i=1}^N$, 我们首先对每个文档 d_i 应用一个经过指令微调的大型语言模型, 并使用以实体为中心的提示, 获得正则实体 \mathcal{E}_i 。随后, 我们将 (d_i, \mathcal{E}_i) 输入到第二个提示中进行知识原子与三元组的联合抽取, 生成原子 $\mathcal{A}_i = \{a_{i,1}, \dots, a_{i,n_i}\}$ 和三元组 $\mathcal{T}_i \subseteq \mathcal{E}_i \times \mathcal{R} \times \mathcal{E}_i$, 其中 \mathcal{R} 表示文本关系标签。三元组仅用于推导图中的辅助实体-实体边; 它们从不作为证据单元直接检索。该知识图谱包含两种结点类型。

知识原子结点。 一个知识原子 $a_{i,j} \in \mathcal{A}_i$ 是从 d_i 中解析出的最小且自包含的自然语言声明。它被编写为上下文完整的、非回指性的, 即避免使用未明确指代的代词 (如 “它”、“他们”), 并能在孤立状态下保持可理解性。在实践中, 一个原子通常聚焦于单一主题或事实 (例如基底细胞皮肤癌的症状), 而不是纠缠多个方面。每个原子都可以作为独立的证据单元进行检索, 并标注其所涉及的实体, 用 $\mathcal{E}(a_{i,j}) \subseteq \mathcal{E}_i$ 表示。

实体结点。 实体 $\mathcal{E}_i = \{e_{i,1}, \dots, e_{i,m_i}\}$ 是从 d_i 中的表面提及 (例如别名或共指提及) 聚合得到的正则概念。

对语料库进行聚合可得到全局词表 $\mathcal{A} = \bigcup_{i=1}^N \mathcal{A}_i$, $\mathcal{E} = \bigcup_{i=1}^N \mathcal{E}_i$, 和 $\mathcal{T} = \bigcup_{i=1}^N \mathcal{T}_i$ 。

连通性组织。 我们将原子-实体图定义为一种异构加权图 $G = (V, \mathcal{L})$, 包含两种结点类型 (原子和实

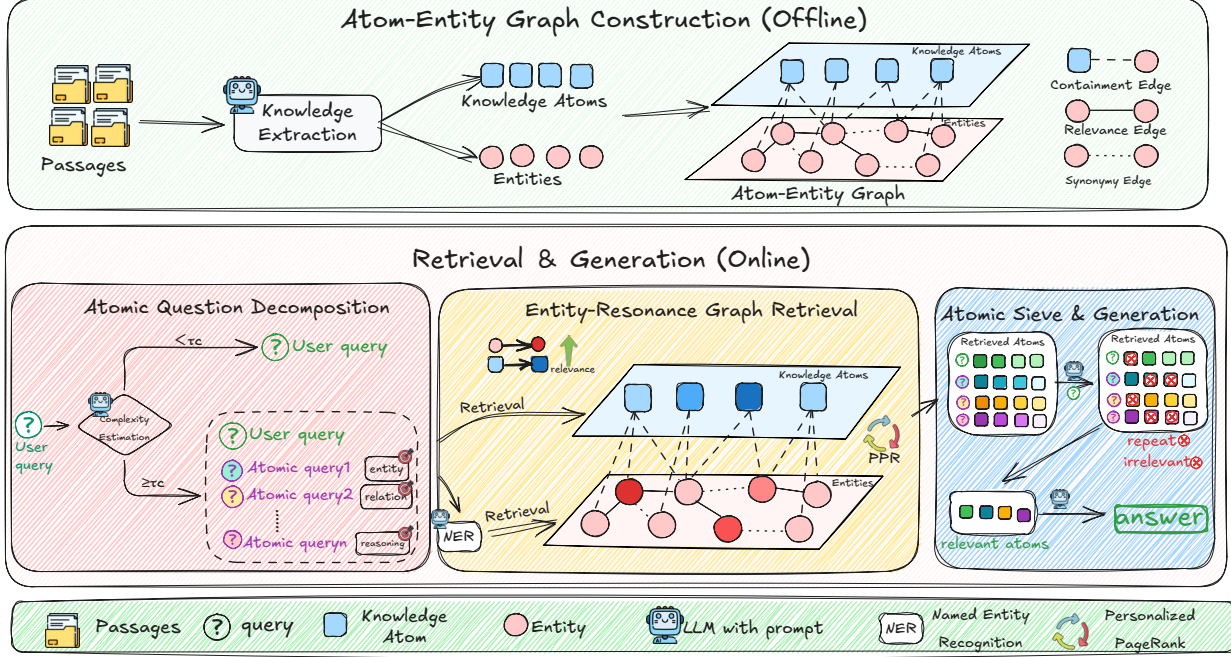


Figure 2. AtomicRAG 概述。在预处理阶段，我们构建一个未标记的原子-实体图（AEG），将语料库分解为通过实体和共现关系连接的极小知识单元。具体而言，如图所示，我们的共现关系分为三类：包含关系、相关性以及多词一义性。在检索阶段，复杂查询可选地被分解为原子子查询，这些子查询会触发在 AEG 上的实体共振传播，以检索多跳证据。最后，一个原子筛滤器对检索到的知识单元进行过滤和合并，生成紧凑且去重的上下文，用于生成有依据的答案。

体) 以及三种边类型。我们有意省略了文本谓词标签，而是将标量权重附加到边上。在传播过程中， \mathcal{L} 中的边被视为双向的。我们通过三种共现关系来组织连通性：包含边、相关边和同义边。

包含边（权重 1）。我们将每个原子与其提及的实体相连，权重为单位权重：

$$\mathcal{L}_{\text{cont}} = \{(a, e) \mid a \in \mathcal{A}, e \in \mathcal{E}(a)\}, \quad w(a, e) = 1. \quad (1)$$

相关边（权重 = # 不同关系类型）。为了在不依赖脆弱的谓词语义的情况下增强跨句子和跨文档的连通性，我们从三元组中推导出实体之间的相关边。对于每对实体 (e, e') ，我们分配一个标量相关权重，该权重等于连接它们的不同关系标签的数量：

$$w(e, e') = \left| \{r \mid (e, r, e') \in \mathcal{T} \text{ or } (e', r, e) \in \mathcal{T}\} \right|. \quad (2)$$

每当 $w(e, e') > 0$ ，我们就添加一条无向相关性边 (e, e') 。我们不存储文本标签 r 在图中；仅保留标量权重 $w(e, e')$ 。

同义边（权重 = 相似度）。为缓解由别名或近义形式引起的碎片化问题，我们连接具有相似表示的实体。令 \mathbf{z}_e 表示由我们的共享编码器（在下一段中详细说明）生成的实体 e 的嵌入。若 $\cos(\mathbf{z}_e, \mathbf{z}_{e'}) \geq \tau_s$ ，则添加一条同义边 (e, e') ，其权重为 $w(e, e') = \cos(\mathbf{z}_e, \mathbf{z}_{e'})$ ：

$$\mathcal{L}_{\text{syn}} = \{(e, e') \mid e, e' \in \mathcal{E}, \cos(\mathbf{z}_e, \mathbf{z}_{e'}) \geq \tau_s\}. \quad (3)$$

最终的边集是 $\mathcal{L} = \mathcal{L}_{\text{cont}} \cup \mathcal{L}_{\text{rel}} \cup \mathcal{L}_{\text{syn}}$ ，其中 $\mathcal{L}_{\text{rel}} = \{(e, e') \mid e, e' \in \mathcal{E}, w(e, e') > 0\}$ 。

向量表示存储。原子、实体和(子)查询通过一个共同的编码器 $f_\theta(\cdot)$ 嵌入到共享的向量空间中，即， $\mathbf{z}_a = f_\theta(a)$ ， $\mathbf{z}_e = f_\theta(e)$ ，以及 $\mathbf{z}_{q'} = f_\theta(q')$ 。在实际应用中，我们存储 (i) 基于原子嵌入的近似最近邻索引以支持语义检索，(ii) 实体的嵌入表，以及 (iii) 由 \mathcal{L} 所诱导的稀疏加权邻接关系，用于图传播。因此，每个知识原子结点同时是一个极小的语义单元，也是与 AEG 相连接的结构钩子。**命题 1.** 原子-实体图提供了更全面且更鲁棒的知识表征。

证明. 我们在第 4.4 节提供了实验证据, 在附录 C.1 中给出了形式化证明。□

3.2. 原子问题分解

复杂查询通常隐式地分解为多个子问题, 其答案依赖于不同的证据片段。将此类查询视为单一检索单元, 迫使检索器匹配一个纠缠的信号, 从而放大语义漂移和检索噪声。为缓解这种不匹配, 我们可选择执行原子问题分解, 生成粒度更符合我们原子知识的查询单元。通过这一分解, 原子子查询与知识原子在证据需求层面实现对齐: 每个 $q^{(t)}$ 均旨在寻找单一、自包含的原子证据单元, 从而在检索过程中减少跨维度的纠缠。

给定一个查询 q , 我们使用一种类似评分标准的指令提示大语言模型 (LLM) 为该查询分配一个结构复杂度得分 $c(q) \in [0, 10]$; 评分提示的详细信息见附录。若 $c(q)$ 超过一个固定阈值 τ_c , 则将该查询分解为一组少量的基本子查询 $\{q^{(1)}, \dots, q^{(m)}\}$, 其中 $m \leq m_{\max}$ 代表子查询的数量。我们明确指示 LLM 生成的子查询应分别针对 q 的特定方面 (例如, 确定实体提及、明确关系或解决中间推理步骤), 而非生成任意的改写表达。随后, 我们定义有效查询集合

$$\tilde{Q}(q) = \begin{cases} \{q\} \cup \{q^{(1)}, \dots, q^{(m)}\}, & c(q) \geq \tau_c, \\ \{q\}, & \text{otherwise.} \end{cases} \quad (4)$$

每个 $q' \in \tilde{Q}(q)$ 在后续的检索阶段中独立处理, 这减少了不同证据需求之间的早期纠缠。

命题 2. 查询与原子知识之间的粒度对齐提升检索效率。

证明. 我们在附录 C.2 中提供了正式证明。□

3.3. 实体共振图检索

对原子进行纯稠密检索缺乏组织多跳证据的机制, 而对噪声丰富的谓词类型关系进行显式符号推理则容易失效。实体共振图检索则利用 AEG 作为未标记的骨架: 它通过共享实体软性传播查询信号, 从而在不依赖语义谓词的情况下, 生成可解释的证据

链。

查询相关个性化. 对于每个有效查询 $q' \in \tilde{Q}(q)$, 我们通过结合两种信号, 在图结点上初始化一个个性化分布: (i) q' 与从原子索引获得的原子嵌入之间的稠密相似度; 以及 (ii) 从 q' 中提取的实体提及, 并映射到实体结点。令 $r_{\text{atom}}^{(0)}(q', v)$ 和 $r_{\text{ent}}^{(0)}(q', v)$ 为在 V 上的非负种子权重, 其中 $r_{\text{atom}}^{(0)}(q', v) = 0$ 对应 $v \notin \mathcal{A}$, $r_{\text{ent}}^{(0)}(q', v) = 0$ 对应 $v \notin \mathcal{E}$ 。我们通过一个标量 α 衰减原子种子, 然后对组合得分进行归一化:

$$\tilde{\pi}_{q'}(v) = \alpha r_{\text{atom}}^{(0)}(q', v) + r_{\text{ent}}^{(0)}(q', v), \quad (5)$$

$$\pi_{q'}(v) = \frac{\tilde{\pi}_{q'}(v)}{\sum_{u \in V} \tilde{\pi}_{q'}(u)}, \quad \sum_{v \in V} \pi_{q'}(v) = 1. \quad (6)$$

此处 α 降低了原子级别相似度相对于基于实体信号的权重; 在所有实验中, 我们设置 $\alpha = 0.1$, 这使得初始化偏向于实体, 同时保留少量原子证据。

共振传播. 令 P 为 G 的行规范化状态转移矩阵。我们计算一个个性化 PageRank 向量 $\mathbf{r}_{q'}$ 作为

$$\mathbf{r}_{q'} = \rho \boldsymbol{\pi}_{q'} + (1 - \rho) P^\top \mathbf{r}_{q'}, \quad (7)$$

其中 $\boldsymbol{\pi}_{q'}$ 是 $\pi_{q'}$ 的向量形式, $\rho \in (0, 1)$ 是重启概率。我们全程设定 $\rho = 0.3$ 。此传播过程沿原子—实体—原子路径分配相关性质量, 放大那些在 q' 中提及 (或通过辅助链接解析) 的实体所结构上支持良好的原子。原子相关性得分直接由

$$s_{q'}(a) = r_{q'}(a), \quad a \in \mathcal{A}, \quad (8)$$

在 G 中的高质路径构成 实体共振链, 为 q' 提供了证据流动的明确解释。

3.4. 原子筛与接地生成

基于图形的 AEG 传播仍可能揭示出松散相关或冗余的原子。因此, 我们采用最终的语义过滤步骤, 在原子层面确保准确率, 而无需退回到粗粒度的检索单元。

原子过滤. 对于每个有效查询 $q' \in \tilde{Q}(q)$, 我们首先根据其共振得分选择一个较小的原子候选集:

$$\mathcal{R}(q') = \text{TopK}_{a \in \mathcal{A}} s_{q'}(a), \quad |\mathcal{R}(q')| = K, \quad (9)$$

在所有实验中使用 $K = 25$ 。原始查询和所有子查询的候选项合并为

$$\mathcal{R}(q) = \bigcup_{q' \in \tilde{\mathcal{Q}}(q)} \mathcal{R}(q'). \quad (10)$$

随后，我们通过提示一个经过指令微调的大型语言模型，对每个 $a \in \mathcal{R}(q)$ 评判 a 是否对原始查询 q 必要且相关，从而获得一个过滤后的子集 $\mathcal{S}(q) \subseteq \mathcal{R}(q)$ 。因此，子查询 q' 仅用于暴露多样化的候选项，而所有包含决策均基于原始信息需求。

聚合与生成。经过筛选的原子单元在源文档层面进一步合并，形成最终的证据集

$$\mathcal{A}^*(q) \subseteq \mathcal{S}(q), \quad (11)$$

将来自同一文档中重叠范围的原子合并为单一引用单元，以避免文本冗余并保持上下文紧凑。

4. 实验

本节介绍了实验设置、主要结果及分析。我们回答以下研究问题 (RQs): **RQ1:** AtomicRAG 是否优于现有方法? **RQ2:** AtomicRAG 的各个主要组件对性能有何贡献? **RQ3:** 我们的原子-实体图是否优于其他图结构组织方式? **RQ4:** 实体共振图检索策略能否提升检索准确率和效率? **RQ5:** AtomicRAG 在索引和生成方面的开销如何? 附加分析详见附录。

4.1. 实验设置

数据集与指标。我们在两个来自 Graph-Bench (Xiang et al., 2025) 的领域特定基准以及三个广泛使用的多跳问答数据集 (HotpotQA (Yang et al., 2018)、2WikiMultiHopQA (Ho et al., 2020) 和 MuSiQue (Trivedi et al., 2021)) 上评估 AtomicRAG 的有效性。对于 Graph-Bench *Medical* 和 *Novel*, 查询被分为四类难度递增的问题类型: 事实检索、复杂推理、上下文摘要和 创造性生成。对于全部五个数据集, 我们遵循 Graph-Bench 的预处理协议以保证一致性: 将文档分割为每段 256 个 token, 重叠部分为 32 个 token。作为评估指标, 我们采用 Graph-Bench 提出的 答案准确率 (ACC), 该指标结合了

基于大语言模型的判断与基于嵌入的语义匹配; 详细的定义与实现方法见附录 A.1.1。

基准与实现细节。我们将基准方法分为两类。(i) 原始 RAG: 使用相同生成器的标准稠密检索流水线, 分别在无重排序和有重排序的情况下进行评估。(ii) 图增强 RAG: 通过显式结构组织证据的代表性系统, 包括 MS-GraphRAG (Edge et al., 2024)、RAPTOR (Sarathi et al., 2024)、LightRAG (Guo et al., 2024)、HippoRAG (Gutierrez et al., 2024)、HippoRAG2 (Gutiérrez et al., 2025)、Fast-GraphRAG (CircleMind-AI, 2024)、LazyGraphRAG (Darren Edge, 2024)、KET-RAG (Huang et al., 2025)、KGP (Wang et al., 2023)、StructRAG (Li et al., 2024) 以及 GFM-RAG (Luo et al., 2025c)。为确保可控比较, 所有方法均采用相同的嵌入模型 (BAAI/bge-large-en-v1.5)。对于答案生成和基于大语言模型的评估, 我们使用相同的骨干大模型 (GPT-4o-mini)。完整规格详见附录 A.1.5。

4.2. 主要结果 (RQ1)

总体比较。为了评估我们方法的有效性, 我们在 Graph-Bench 和多跳问答基准上, 将该方法与多种强大的原始 RAG 变体以及广泛的图增强 RAG 基准进行了对比。结果如表 1 所示。我们的方法在所有任务列的总体平均得分 (Avg.=64.9) 上表现最佳, 始终优于所有基线方法。值得注意的是, 我们在大多数任务列中取得了最高得分, 并在 Graph-Bench (Novel) *Reason* 上并列第一, 表明性能提升并非由单一数据集或问题类型驱动, 而是在各类任务中均具有普遍性。

在多个基准和领域上。性能提升在所有基准组中保持一致。在 Graph-Bench (医学) 上, 我们达到 73.1 平均分, 相较于最佳基准提升 +8.3; 在 Graph-Bench (新领域) 上, 我们获得 60.7 平均分, 提升 +4.3; 在多跳问答任务中, 我们取得 59.4 平均分, 提升 +6.0。在更难的基准和需要串联分散证据的领域 (例如 MuSiQue) 中, 提升幅度最大, 这一模式表明性能提升源于多跳证据组合能力的增强, 而非仅限于

Table 1. 在 Graph-Bench 和多跳问答基准上的性能比较。Fact、Reason、Summ. 和 Creat. 分别表示事实检索、复杂推理、上下文摘要和创意生成。最终 Avg. 为所有任务的平均得分。最佳结果以 **粗体** 标出，次佳结果以 下划线 标出。提升行报告了 Ours 相对于最佳基线的绝对得分提升（以分计）；↑ 表示提升。

Method	Graph-Bench (Medical)					Graph-Bench (Novel)					Multi-hop QA				Avg.
	Fact	Reason	Summ.	Creat.	Avg.	Fact	Reason	Summ.	Creat.	Avg.	HotpotQA	2Wiki	MuSiQue	Avg.	
Vanilla Retrieval-Augmented Generation															
RAG (w/o rerank)	63.7	57.6	63.7	58.9	61.0	58.7	41.4	50.1	41.5	47.9	54.0	31.8	32.2	39.3	50.3
RAG (w rerank)	64.7	58.6	65.8	60.6	62.4	<u>60.9</u>	42.9	51.3	38.3	48.3	54.5	32.1	33.3	40.0	51.2
Graph-based Retrieval-Augmented Generation															
MS-GraphRAG (local)	38.6	47.0	41.8	53.1	45.1	49.3	<u>50.9</u>	64.4	39.1	50.9	52.2	37.4	37.8	42.5	46.5
MS-GraphRAG (global)	16.4	15.6	19.8	20.8	18.1	36.9	43.2	56.9	41.1	44.5	37.6	39.5	35.4	37.5	33.0
HippoRAG	56.1	55.8	59.8	64.4	59.0	52.9	38.5	48.7	38.9	44.8	38.1	29.8	25.7	31.2	46.2
HippoRAG2	<u>66.3</u>	61.9	63.1	<u>68.1</u>	<u>64.8</u>	60.1	53.4	64.1	48.3	<u>56.5</u>	67.5	47.5	41.5	52.2	<u>58.3</u>
LightRAG	63.3	61.3	63.1	67.9	63.9	58.6	49.1	48.9	23.8	45.1	57.0	44.2	23.5	41.6	51.0
Fast-GraphRAG	60.9	61.7	<u>67.9</u>	65.9	64.1	56.9	48.5	56.4	46.2	52.0	62.3	47.8	42.1	50.7	56.1
RAPTOR	54.0	53.2	58.7	62.4	57.1	49.3	38.6	47.1	38.0	43.2	66.4	50.1	<u>43.6</u>	<u>53.4</u>	51.0
Lazy-GraphRAG	60.3	47.8	57.3	62.2	56.9	51.7	49.2	58.3	43.2	50.6	54.6	40.4	39.1	44.7	51.3
KGP	52.3	51.5	54.5	63.8	55.5	54.2	46.3	51.2	40.3	48.0	52.3	38.6	37.2	42.7	49.3
StructRAG	55.4	56.2	62.5	60.2	58.6	53.8	46.3	54.3	42.2	49.1	49.2	<u>52.3</u>	24.9	42.1	50.7
KET-RAG	60.4	39.6	45.3	43.0	47.1	55.4	36.6	52.5	46.0	47.6	45.6	24.5	22.6	30.9	42.9
GFM-RAG	63.5	<u>67.3</u>	51.5	63.3	61.4	51.0	49.8	<u>66.3</u>	<u>57.8</u>	56.2	<u>68.6</u>	43.6	39.9	50.7	56.6
Ours	72.6	74.8	76.8	68.3	73.1	61.0	53.4	68.5	60.0	60.7	70.5	56.8	50.9	59.4	64.9
Improv. vs best baseline	↑6.3	↑7.5	↑8.9	↑0.2	↑8.3	↑0.1	-	↑2.2	↑2.2	↑4.3	↑1.9	↑4.5	↑7.3	↑6.0	↑6.6

单跳匹配的改进。

跨问题类型。我们的方法在各类问题上均实现了统一的性能提升，体现了广泛的覆盖性，而非孤立的、特定类型的优势。在 Graph-Bench（医学）上，我们同时将 事实/推理/总结 的性能提升了 +6.3/+7.5/+8.9，表明相同的设计选择增强了事实基础、多步证据链以及跨原子合成能力，而非过度优化单一技能。在 Graph-Bench（新领域）上，我们在原本表现强劲的类型中保持竞争力（通常为最佳或接近最佳），同时仍提升了较弱类型的性能，从而缩小了各类别之间的差距，并提高了整体上限。相比之下，多个基准方法在不同问题类型间表现出更高的方差——在部分类型上表现优异，但在其他类型上性能下降——而我们的方法在所有类别中始终保持稳健，显示出对问题风格变化更强的鲁棒性。

Table 2. 在三个数据集上的消融实验。括号内表示相对于 AtomicRAG 的得分下降。ERGR：实体共振图检索；AQD：原子问题分解；AS：原子筛选；KA：知识原子化。

Method	HotpotQA	Medical	Novel	Avg.
AtomicRAG	70.5	73.2	60.7	68.1
Single-module removal				
w/o ERGR	68.4(↓2.1)	72.0(↓1.2)	59.1(↓1.6)	66.5(↓1.6)
w/o AQD	67.8(↓2.7)	<u>72.3</u> (↓0.9)	<u>60.0</u> (↓0.7)	66.7(↓1.4)
w/o AS	68.0(↓2.5)	71.5(↓1.7)	59.8(↓0.9)	66.4(↓1.7)
w/o KA	62.2(↓8.3)	64.3(↓8.9)	52.7(↓8.0)	59.7(↓8.4)
Combined removal				
w/o {ERGR, AQD}	66.0(↓4.5)	70.8(↓2.4)	58.6(↓2.1)	65.1(↓3.0)
w/o {ERGR, AQD, AS}	64.5(↓6.0)	69.5(↓3.7)	57.5(↓3.2)	63.8(↓4.3)
w/o {ERGR, AQD, AS, KA}	54.0(↓16.5)	61.0(↓12.2)	47.9(↓12.8)	54.3(↓13.8)

4.3. 消融实验结果 (RQ2)

表 2 报告了在 HotpotQA、Graph-Bench (Medical) 和 Graph-Bench (Novel) 上的消融实验，使用 Avg. 作为主要指标。AtomicRAG 达到了 68.1 的 Avg. 移除任意单一模块都会导致性能下降，这表明性能提升来自各组件之间的互补作用，而非单一组件。

Table 3. 图结构统计量。我们报告了在 Graph-Bench (医学) 上构建的图的结点数、边数、平均度和平均聚类系数。

Method	#Nodes	#Edges	Avg. degree	Avg. clustering
KET-RAG	3,134	2,421	1.55	0.24
LightRAG	1,942	2,220	2.29	0.09
HippoRAG2	10,660	119,489	22.42	0.38
GFM-RAG	9,569	40,444	8.45	0.29
Ours	14,586	146,115	23.04	0.39

单模块影响。每个模块均带来可衡量的收益：移除 ERGR/AQD/AS 分别使平均值下降 1.6/1.4/1.7, 而移除 KA 导致最大单模块下降至 59.7 (−8.4)。这些影响与其功能一致：AQD 在 HotpotQA 上最为关键 (−2.7)，ERGR 在各数据集上均导致性能均匀下降，而移除 AS 始终损害性能，表明筛选能有效过滤噪声原子并提升证据准确率。

联合移除下的协同作用。当模块联合移除时，性能下降幅度叠加：同时移除 {ERGR, AQD} 会使平均值下降 3.0，进一步移除 AS 会使下降幅度增加至 4.3。这表明分解 (AQD)、图检索 (ERGR) 和最终过滤 (AS) 之间存在明显的互补性。

知识原子化是基础性的。移除知识原子化 (KA) 会导致性能显著下降。在 *w/o KA* 变体中，我们禁用知识原子化，并将原子知识单元替换为原始文本块作为检索单元。即使仅移除 KA，平均得分也急剧下降至 59.7 (−8.4)，而进一步移除所有模块 (包括 KA) 后，性能进一步降至 54.3 (−13.8)。这些结果表明，原子级别的粒度至关重要：缺乏该粒度时，证据路径的构建与优化无法可靠进行，系统几乎退化为粗粒度的文本块检索。

4.4. 图质量分析 (RQ3)

接下来，我们从结构连通性和语义实用性两个方面考察所构建图的质量。

结构连通性。表 3 报告了基本的结构统计量。与先前的基准方法相比，我们的图更大，并表现出稍强的局部连通性。这种连通性有利于组合多跳证据，但本身并不能保证邻域是正确或对定位有用的。

语义效用。为评估邻域是否具有语义帮助性，我们进

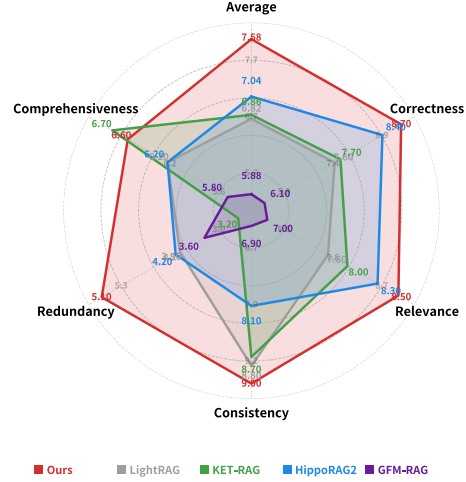


Figure 3. 语义效用。基于大语言模型对 Graph-Bench (医学) 数据集上 1 跳图邻域的评估，涵盖正确性、相关性、一致性、冗余性和全面性。

行基于大语言模型的邻域评估。在 Graph-Bench (医学) 上，我们选取 10 个高频实体 (根据语料库提及次数)，并从每个构建的图中提取其 1 跳邻域。给定中心实体及其邻域，我们向 gpt-oss-120b (Agarwal et al., 2025) 提供提示，从五个维度对邻域整体进行评分：正确性、相关性、一致性、冗余度和全面性。我们将各实体的得分取平均，并在图 3 中可视化结果。我们观察到，与 KET-RAG 相比，全面性略有下降 (6.60 对比 6.70)，而 KET-RAG 在冗余度上的得分显著更低。总体而言，AtomicRAG 在正确性、相关性和一致性方面表现更优，同时降低了冗余度，从而为 RAG 提供了更具实用性的知识锚定邻域。

4.5. 检索效率分析 (RQ4)

我们从以下三个方面评估效率：(i) 随着 Top-*k* 变化时的准确率–token 权衡，(ii) 在固定上下文预算下的鲁棒性，以及 (iii) 每次查询的检索延迟。

Top-*k* 对效果的影响。在图 4 中，token 消耗量从 $k=1$ 时的 3245 单调增加至 $k=25$ 时的 6230，而准确率在较小的 k 时迅速提升 (例如，*Reason* 和 *Summ.* 从 $k=1$ 到 $k=3$ 显著上升)，随后在较大的 k 时趋于饱和 (大约在 $k \geq 10$ 附近)，表明大多数决定性证据已在适度的 Top-*k* 范围内覆盖，而额外的检索主

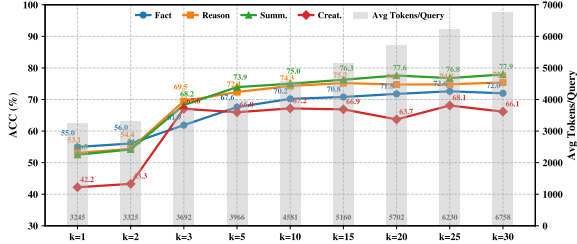


Figure 4. 检索 Top- k 超参数对答案准确率和 token 长度的影响: Top- k 指定 AtomicRAG 每次查询检索的知识原子数量, 而 token 长度是指由问题和检索到的原子组成的 LLM 输入中的总 token 数。

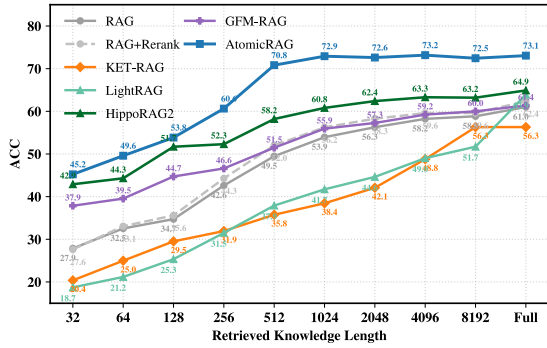


Figure 5. 有限上下文长度下的准确率: 每个点均在固定的上下文预算下进行评估, 该预算定义为大语言模型输入中允许的最大 token 数量, 所有方法在生成前均被截断至该预算。

要引入了冗余。

有限长度下的性能。图 5 显示, 在严格的预算条件下, 我们的方法仍保持强劲表现, 在上下文预算为 512 token 时达到 70.8 的准确率, 并在 73 左右趋于稳定, 而强基准方法提升较为缓慢, 需要显著更长的上下文才能接近其最佳性能, 表明在长度受限的情景下, 我们检索到的证据更加稠密且更高效利用预算。

检索延迟。表 4 报告了每次查询的检索延迟。我们的方法在基于图形的基准中速度最快, 每次查询仅需 0.79 秒, 而 KET-RAG 为 2.73 秒, HippoRAG2 为 4.50 秒, GFM-RAG 为 6.08 秒, LightRAG 为 13.99 秒。这一加速效果与我们的原子-实体表示一致: 相关性传播在紧凑图上进行, 避免了先验系统中重复的多轮扩展。

Table 4. 效率与性能对比。我们报告索引和问答 (QA) 的 token 消耗量 (以百万计, M)、总体 token 成本 (索引 + QA)、平均检索延迟以及答案准确率 (ACC)。每个 token 条目均以总计展示, 并以灰色显示 (提示 + 完成) 的分解。

Method	Index tokens (M)	QA tokens (M)	Total tokens (M)	Latency (s/q)	ACC (%)
Vanilla RAG	0.00 (0.00 + 0.00)	2.87 (2.68 + 0.19)	2.87	0.01	62.4
KET-RAG	2.31 (1.74 + 0.57)	24.45 (24.42 + 0.03)	26.76	2.73	47.1
LightRAG	2.66 (2.35 + 0.31)	61.51 (60.74 + 0.77)	64.17	13.99	63.9
HippoRAG2	1.54 (1.18 + 0.36)	4.29 (3.96 + 0.34)	5.83	4.50	64.9
GFM-RAG	1.52 (1.19 + 0.33)	2.08 (1.72 + 0.36)	3.60	6.08	61.4
Ours	2.24 (1.67 + 0.56)	2.63 (1.97 + 0.67)	4.87	0.79	73.2

4.6. Token 开销 (RQ5)

表 4 报告了在 Graph-Bench (Medical) 上的索引/查询 token 使用量、延迟和准确率。我们的方法在保持低查询时 token 消耗方面优于那些需要昂贵多轮扩展的图方法。尽管 GFM-RAG 的总 token 数较少 (3.60M), 但其速度较慢 (6.08 秒/查询) 且准确率较低 (61.4 ACC); AtomicRAG 在保证低延迟的同时实现了最佳准确率, 且总 token 数处于合理水平。与 HippoRAG2 相比, 我们在索引阶段多使用了少量 token (+0.70M), 但在查询阶段节省了更多 token (-1.66M), 最终总 token 成本更低 (4.87M 对比 5.83M), 同时准确率提升了 8.3 个百分点。总体而言, AtomicRAG 在图构建阶段承担了较小的额外开销, 但该开销通过降低查询阶段的 token 使用量以及提升答案准确率得到了充分补偿。

5. 结论

本研究揭示了现有 RAG 系统中知识表征与知识索引之间存在的核心不匹配问题, 并提出 AtomicRAG 以显式地解耦二者: 语义内容仅由知识原子承载, 而未标记的原子-实体图仅提供可达性与聚合先验, 而非编码谓词语义。大量实验表明, 该设计能够生成更稳定的证据链、更紧凑的检索上下文, 并在多跳情景下实现更好的准确率-效率权衡, 使 AtomicRAG 成为应对知识密集型复杂查询的实用解决方案。

影响声明

本文介绍了旨在推动机器学习领域发展的研究工作。我们的研究可能带来多种社会影响, 但均无需在此特别强调。

参考文献

- Agarwal, O. S. et al. gpt-oss-120b&gpt-oss-20b model card. 2025.
- Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *ArXiv*, abs/2310.11511, 2023.
- Chen, S., Zhou, C., Yuan, Z., Zhang, Q., Cui, Z., Chen, H., Xiao, Y., Cao, J., and Huang, X. You don’t need pre-built graphs for rag: Retrieval augmented generation with adaptive reasoning structures. *ArXiv*, abs/2508.06105, 2025.
- Chen, T., Wang, H., Chen, S., Yu, W., Ma, K., Zhao, X., Yu, D., and Zhang, H. Dense x retrieval: What retrieval granularity should we use? In *Conference on Empirical Methods in Natural Language Processing*, 2023.
- CircleMind-AI. Fastgraphrag: High-speed graph-based retrieval-augmented generation. *CircleMind-AI Blog*, 2024.
- Darren Edge, Ha Trinh, J. L. Lazygraphrag: Setting a new standard for quality and cost. *Microsoft Blog*, 2024.
- Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A. N., Truitt, S., and Larson, J. From local to global: A graph rag approach to query-focused summarization. *ArXiv*, abs/2404.16130, 2024.
- Gao, L., Ma, X., Lin, J. J., and Callan, J. Precise zero-shot dense retrieval without relevance labels. In *Annual Meeting of the Association for Computational Linguistics*, 2022.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., and Wang, H. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997, 2023.
- Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.
- Guo, Y., Su, M., Guan, S., Sun, Z., Jin, X., Guo, J., and Cheng, X. Routerag: Efficient retrieval-augmented generation from text and graph via reinforcement learning. 2025b.
- Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. Lightrag: Simple and fast retrieval-augmented generation. *ArXiv*, abs/2410.05779, 2024.
- Gutierrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., and Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. *ArXiv*, abs/2405.14831, 2024.
- Guti’errez, B. J., Shu, Y., Qi, W., Zhou, S., and Su, Y. From rag to memory: Non-parametric continual learning for large language models. *ArXiv*, abs/2502.14802, 2025.
- Ho, X., Nguyen, A., Sugawara, S., and Aizawa, A. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *ArXiv*, abs/2011.01060, 2020.
- Hou, Y., Zhou, S., Liang, K., Meng, L., Chen, X., Xu, K., Wang, S., Liu, X., and Huang, J. Soft reasoning paths for knowledge graph completion. In *International Joint Conference on Artificial Intelligence*, 2025.
- Hu, Y., Lei, Z., Zhang, Z., Pan, B., Ling, C., and Zhao, L. Grag: Graph retrieval-augmented generation. *ArXiv*, abs/2405.16506, 2024.

- Huang, Y., Zhang, S., and Xiao, X. Ket-rag: A cost-efficient multi-granular indexing framework for graph-rag. *arXiv preprint arXiv:2502.09304*, 2025.
- Izacard, G. and Grave, E. Leveraging passage retrieval with generative models for open domain question answering. *ArXiv*, abs/2007.01282, 2020.
- Jiang, H., Wu, Q., Lin, C.-Y., Yang, Y., and Qiu, L. LlmLingua: Compressing prompts for accelerated inference of large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.
- Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L. Y., Edunov, S., Chen, D., and tau Yih, W. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906, 2020.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020.
- Li, Z., Chen, X., Yu, H., Lin, H., Lu, Y., Tang, Q., Huang, F., Han, X., Sun, L., and Li, Y. Struc-trag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *ArXiv*, abs/2410.08815, 2024.
- Luo, H., Chen, G., Zheng, Y., Wu, X., Guo, Y., Lin, Q., Feng, Y., Kuang, Z., Song, M., Zhu, Y., et al. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *arXiv preprint arXiv:2503.21322*, 2025a.
- Luo, H., Haihong, E., Chen, G., Lin, Q., Guo, Y., Xu, F., min Kuang, Z., Song, M., Wu, X., Zhu, Y., and Luu, A. T. Graph-r1: Towards agentic graphrag framework via end-to-end reinforcement learning. *ArXiv*, abs/2507.21892, 2025b.
- Luo, L., Zhao, Z., Haffari, G., Phung, D., Gong, C., and Pan, S. Gfm-rag: Graph foundation model for retrieval augmented generation. *ArXiv*, abs/2502.01113, 2025c.
- Mavromatis, C. and Karypis, G. Gnn-rag: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Annual Meeting of the Association for Computational Linguistics*, 2025.
- Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., and Tang, S. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 2024.
- Qian, H., Zhang, P., Liu, Z., Mao, K., and Dou, Z. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*, 2024.
- Sarathi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., and Manning, C. D. Raptor: Recursive abstractive processing for tree-organized retrieval. *ArXiv*, abs/2401.18059, 2024.
- Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., and Chen, W. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *ArXiv*, abs/2305.15294, 2023.
- Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2021.
- Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive

multi-step questions. *ArXiv*, abs/2212.10509, 2022.

Wang, S., Yang, H., and Liu, W. Research on the construction and application of retrieval enhanced generation (rag) model based on knowledge graph. *Scientific Reports*, 15, 2025.

Wang, Y., Lipka, N., Rossi, R. A., Siu, A. F., Zhang, R., and Derr, T. Knowledge graph prompting for multi-document question answering. In *AAAI Conference on Artificial Intelligence*, 2023.

Xiang, Z., Wu, C., Zhang, Q., Chen, S., Hong, Z., Huang, X., and Su, J. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *ArXiv*, abs/2506.05690, 2025.

Xiong, W., Li, X. L., Iyer, S., Du, J., Lewis, P., Wang, W. Y., Mehdad, Y., tau Yih, W., Riedel, S., Kiela, D., and Oğuz, B. Answering complex open-domain questions with multi-hop dense retrieval. *ArXiv*, abs/2009.12756, 2020.

Yang, Q. A., Yang, B., Zhang, B., et al. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. 2018.

Zhang, Q., Chen, S., Bei, Y.-Q., Yuan, Z., Zhou, H., Hong, Z., Dong, J., Chen, H., Chang, Y., and Huang, X. A survey of graph retrieval-augmented generation for customized large language models. *ArXiv*, abs/2501.13958, 2025.

附录

附录提供了补充材料，以补充主论文内容，实现完全可复现性，隔离性能提升的来源，并提供理论和定性洞察。第 A 节记录了实现细节和评估协议，包括基准配置、提示模板、各数据集的图统计量以及运行时间/token/成本分解，确保所有结果均能在一致的情景下复现。第 B 节报告了消融实验和敏感性分析，通过改变嵌入模型、大语言模型骨干网络、图检索策略以及关键的个性化页面排名 (PPR) 超参数，明确哪些设计选择驱动了性能提升。第 C 节给出了主文中所陈述理论命题的完整证明。第 D 节提供了定性案例研究，对比原子级 RAG 与代表性 RAG 变体，展示典型失败模式，并说明原子级别的结构化检索如何缓解这些问题。第 E 节包含流水线中使用的完整提示模板，以实现确切的复现性。最后，第 ?? 节分析了原子级 RAG 的局限性，讨论实际中的失败模式及改进方向。

A. 可复现性详情

A.1. 基准和实现细节

A.1.1. 基准线

我们将基准方法分为两类：(i) **Vanilla RAG**，即采用相同生成器的标准稠密检索流水线，分别在不使用重排序和使用重排序的情况下进行评估；(ii) **Graph-enhanced RAG**，代表性的基于显式结构组织证据的系统，包括 MS-GraphRAG (本地/全局)、RAPTOR、LightRAG、HippoRAG、HippoRAG2、Fast-GraphRAG、LazyGraphRAG、KET-RAG、KGP、StructRAG 以及 GFM-RAG。

香草 RAG。我们在文本块上实现了一个标准的稠密检索器，并使用与 AtomicRAG 相同的生成器。我们报告了两种变体：**无重排序**，采用检索器返回的顺序；以及**带重排序**，在生成前使用 `bge-reranker-large` 对检索到的文本块进行重排序，以改善证据优先级。

猎鹰 RAPTOR 通过递归聚类块并将其汇总为更高级别的结点，构建层次化树状索引。在推理阶段，它可跨多个抽象层级（叶块和内部摘要）进行检索，从而实现兼顾局部细节与全局上下文的证据选择。

MS-GraphRAG (本地/全局) MS-GraphRAG 将语料库组织成实体/社区图，并支持两种检索模式。**局部**检索从以实体为中心的邻域中收集细粒度的支持证据，而**全局**检索则聚合社区级别的证据以回答语料库层面的问题，通常通过结构化聚合和摘要实现。

LightRAG LightRAG 在索引和检索中引入了轻量级图结构组织，支持本地证据查找以及对图结构信息的更高级发现，强调在实际部署中的简洁性和高效性。

HippoRAG HippoRAG 从语料库中构建一个无模式的知识图谱，并通过图传播（例如，个性化 PageRank）实现关联的多跳检索。该机制在无需依赖昂贵的迭代提示或复杂的推理时探索的情况下，提升了多跳证据发现的效果。

HippoRAG2 HippoRAG2 在 HippoRAG 风格的关联检索基础上，增强了段落整合能力，并更有效地利用了大语言模型模块，旨在提升多跳推理情景下的证据连通性与鲁棒性。

快速图 RAG。 Fast-GraphRAG 是一种以效率为导向的 GraphRAG 变体，通过快速图探索/传播来加速结构感知检索，在实际延迟约束下识别相关结点和文本片段。

懒惰图 RAG LazyGraphRAG 通过将部分与结构相关的检索任务推迟到推理阶段，减少了前期的索引和摘要成本，并利用有预算的/即时探索来平衡计算成本与答案质量。

KET-RAG。 KET-RAG 通过多粒度构建实现高性价比的索引：它首先选择关键片段以构建轻量级图骨架，然后利用整个语料库上的辅助结构来支持检索，而无需完全生成稠密知识图谱。

KGP。 知识图谱提示 (KGP) 采用基于大语言模型引导的遍历过程，在段落/结构图上迭代地导航结点，以收集支持性段落。该图为多文档和多跳问答系统中的证据转移提供了全局约束。

StructRAG StructRAG 在推理时执行混合结构化：它先检索原始证据，然后将其重构为与任务相适应的模式或结构化上下文，再进行推理，旨在提升全局整合能力，并降低对零散或嘈杂证据的敏感性。

GFM-RAG。 GFM-RAG 采用学成的图检索器（例如基于图神经网络的检索器）对图结构进行推理并检索相关证据，相较于在噪声图上仅依赖启发式遍历或传播的方法，提升了鲁棒性。

实施与评估协议。 所有图增强基线均严格遵循 GRAPH-BENCH 的配置和评估协议，以确保公平比较。对于 **Medical** 和 **Novel** 基准，我们直接报告官方的 GRAPH-BENCH 结果，因为这些数据集在固定的标准情景下进行评估。对于 **Multi-Hop** 任务，我们根据相应的 GRAPH-BENCH 配置，忠实复现每个基线的索引、检索和推理流程，并在与 AtomicRAG 相同的实验条件下进行评估。

A.1.2. 原子 RAG 默认配置

骨干模型。 我们使用 gpt-4o-mini 作为生成式大模型，BAAI/bge-large-en-v1.5 作为嵌入模型。嵌入的最大序列长度设置为 2048。

我们设置 `retrieval_top_k=25` 用于候选检索。对于多跳查询，我们采用保守的分解预算 (`max_sub_questions=3`)，并提高分解触发阈值 (`complexity_threshold=6.5`)，以避免对简单查询过度碎片化，同时仍能在真正复杂的问答中启用分解。对于图谱检索，我们使用个性化页面排名 (`propagation_method=ppr`)，其中 `damping=0.3`；多词一义性边通过 KNN 构建，`synonymy_edge_topk=2047`，并经由 `synonymy_edge_sim_threshold=0.8` 进行过滤。

A.1.3. 按数据集的图统计量

表 6 总结了每个数据集所构建的原子-实体图 (AEG) 的图级统计量。在我们的 AEG 中，**结点是实体结点和原子知识结点** (知识原子) 的并集。**边**被划分为三种类型：(i) **相关边**连接实体-实体对 (捕捉共现/关联)，(ii) **同义边**连接语义相似的实体 (通过嵌入 KNN 和阈值化构建)，以及 (iii) **包含边**连接提及某实体的原子 (原子-实体关联)。这些统计量提供了语料库依赖的图大小和稀疏性的具体视图，直接影响索引成本和基于传播的检索效率。

在不同数据集上，AEG 的大小随语料库复杂度而变化：多跳问答数据集 (如 HotpotQA 和 MuSiQue) 生成的图谱最大 (结点数和总边数均最多)，反映出更广泛的实体覆盖范围和更强的实体间连接密度。相比之

Component / Parameter	Value	Description
LLM	gpt-4o-mini	Generator used for final answer synthesis.
Embedding model	BAAI/bge-large-en-v1.5	Dense encoder for atoms/passages and queries.
embedding_max_seq_len	2048	Maximum input length for the embedding model.
retrieval_top_k	25	Number of retrieved candidates atoms per query for downstream selection.
synonymy_edge_topk	2047	Top- k nearest neighbors used to construct synonymy edges (KNN).
synonymy_edge_sim_threshold	0.8	Minimum similarity required to add a synonymy edge.
entity_node_weight	1.0	Weight factor for entity seeds when initializing propagation.
entity_top_k	20	Max number of entity nodes retained per query as initial seeds.
entity_sim_threshold	0.3	Minimum similarity for an entity to be considered a valid seed.
propagation_method	ppr	Graph propagation method (Personalized PageRank).
damping	0.3	PPR damping factor controlling restart probability.
passage_node_weight	0.1	Weight assigned to passage/atom nodes in the propagation graph.
propagation_num_iter	20	Iterations for iterative propagation.
propagation_num_walks	1000	Number of random walks used for Monte-Carlo PPR estimation.
propagation_walk_length	10	Length of each random walk.
max_sub_questions	3	Maximum number of induced sub-questions per query .
complexity_threshold	6.5	Threshold for triggering query decomposition.

Table 5. 我们原子 RAG 实现中使用的超参数及默认设置。

下，医学类数据集结点数量显著较少，但每个结点的边数相对较高，表明在同义关系与包含关系下，实体空间具有更高的连接密度。新数据集表现出中等规模的图谱，边的构成较为均衡，这与叙事型语料库一致，这类语料库引入大量实体，但跨文档关联数量相较于百科式问答基准数据集较少。

A.1.4. 运行时、TOKEN 数量及成本分解

表 7 报告了 ATOMICRAG 主要模块在运行时间、token 使用量和货币成本方面的详细分解。结果突显了图中心与大模型中心成本之间的明显分离。

从运行时的角度来看，**实体共振图检索**主导了端到端延迟，占总执行时间的三分之一以上。这反映了大规模图遍历与传播的开销，其计算密集但与大语言模型（LLM）的使用关系不大。相比之下，涉及大量 LLM 交互的模块——如**原子筛检**和**基于事实的答案生成**——尽管进行了大量提示，却仅消耗了较少的墙钟时间。

Dataset	#Nodes	#Entities	#Atoms	#Edges	#Related	#Synonym	#Containment
HotpotQA	112,995	84,592	28,403	479,895	177,734	276,577	25,584
MuSiQue	118,218	87,446	30,772	488,497	175,794	286,923	25,780
2WikiMultiHop	59,782	44,750	15,032	234,008	93,446	125,422	15,140
Medical	14,647	9,087	5,560	143,919	21,068	99,707	23,144
Novel	59,807	40,217	19,590	176,948	75,665	83,481	17,802

Table 6. 每个数据集的原子-实体图 (AEG) 统计量。结点包括实体结点和原子知识结点。边包括实体-实体关联边、同义边以及原子-实体包含边。

Module (AtomicRAG)	Runtime		Tokens				Cost (USD)		
	Time (s)	Share	Prompt	Completion	Total	Share	Prompt	Completion	Total
Atom-Entity Graph Construction	188.84	9.4%	1,673,875	563,101	2,236,976	13.8%	0.251	0.338	0.589
Atomic Question Decomposition	111.60	5.5%	739,474	303,724	1,043,198	6.5%	0.111	0.182	0.293
Entity-Resonance Graph Retrieval	735.80	36.5%	—	—	—	—	—	—	—
Atomic Sieve	210.69	10.4%	10,098,938	158,475	10,257,413	63.4%	1.515	0.095	1.610
Grounded Answer Generation	210.33	10.4%	1,967,021	666,828	2,633,849	16.3%	0.295	0.400	0.695
Total	2,018.46	100%	14,479,308	1,692,128	16,171,436	100%	2.172	1.015	3.187

Table 7. 模块级别的运行时、token 消耗和货币成本。AtomicRAG 中，“Prompt”和“Completion”分别表示底层大语言模型调用的输入和输出 token。成本以美元 (USD) 为单位报告。

从 token 和成本的角度来看，**Atomic Sieve** 是主要贡献者，负责超过总 token 消耗和成本的 60%。这是意料之中的，因为该筛选器在长 prompt 上执行细粒度、片段级别的相关性过滤。Atom-Entity 图构建和原子问题分解会产生中等水平的一次性或查询级 LLM 成本，而 Entity-Resonance 图检索则不引入任何 LLM token 开销。

总体而言，这一分解表明，AtomicRAG 的计算成本主要由少数可解释的阶段主导：延迟方面的图传播以及货币成本方面的基于大模型的原子过滤。这种模块化的分离使得可以有针对性地进行优化——例如加速图遍历，或在筛选前剪枝候选原子——而无需重新设计整个流水线。

A.1.5. 评价指标

我们遵循 Graph-Bench 中的评估协议，并报告标准的生成质量指标。特别地，我们采用 **答案准确率** 作为主要准确率指标，该指标联合评估语义对齐性和事实正确性，以避免过度奖励那些流畅但存在幻觉的答案，或虽事实正确但语义不匹配的答案。值得注意的是，该指标结合了基于大语言模型的声明级验证器与基于嵌入的语义相似度，形成一种混合评分函数，能够反映事实忠实性与语义对齐性（验证器提示模板见 Graph-Bench）。

准确率 答案准确率 (ACC) 通过结合 (i) 语义相似度和 (ii) 细粒度的事实验证，对答案质量进行了双重评估：

$$\text{ACC} = \alpha \cdot \text{FC} + (1 - \alpha) \cdot \text{SS}, \quad (12)$$

其中 α 是一个权重系数（我们默认使用 $\alpha = 0.7$ ，遵循 Graph-Bench）。

Embedding Model	Fact	Reason	Summ.	Creat.	Avg.
bge-large-en-v1.5	72.6	74.8	<u>76.8</u>	68.3	73.1
gte-qwen2 (1.5B)	70.9	<u>74.3</u>	76.2	<u>66.7</u>	<u>72.0</u>
gte-large	<u>71.4</u>	73.7	76.2	66.3	71.9
nomic-embed	71.1	73.4	77.3	65.7	71.9
bge-m3	71.7	74.3	75.1	66.4	71.8
e5-large	70.2	71.5	75.1	66.7	70.9

Table 8. 嵌入模型的消融实验。最佳结果用粗体表示，次佳结果用下划线表示。

事实正确性。 事实正确性 FC 通过逐声明验证计算，并总结为一种 F1 风格的得分：

$$FC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \quad (13)$$

其中，TP 表示已验证的正确命题数量，FP 表示错误（幻觉）命题的数量，FN 表示生成答案未覆盖的参考命题缺失数量。该公式明确对幻觉（假阳性，FP）和遗漏（假阴性，FN）进行惩罚。在 Graph-Bench 中，TP/FP/FN 通过提示大语言模型（LLM）将细粒度命题与参考证据进行验证获得；具体验证指令请参见 Graph-Bench 官方提示模板。

语义相似度 语义相似度 SS 通过基于嵌入的余弦相似度来衡量：

$$SS = \cos(\mathbf{e}(\hat{y}), \mathbf{e}(y)), \quad (14)$$

其中 \hat{y} 和 y 分别表示生成的答案和参考答案， $\mathbf{e}(\cdot)$ 将文本映射到其嵌入表示。该术语即使在表面形式不同时也能奖励语义对齐。

B. 消融分析与敏感性分析

B.1. 嵌入模型消融研究

我们在保持原子-实体图构建、检索和生成提示固定的情况下，对嵌入模型进行了消融实验。表 8 显示，性能对嵌入选择具有中等敏感性：更强的通用英文嵌入能够持续提升事实检索与多跳推理表现，进而提高整体平均得分。在所有候选模型中，bge-large-en-v1.5 在平均得分上表现最佳，并在四个类别中的三个领先，表明我们的检索流水线最受益于在原子文本粒度下具备高语义可分性的嵌入。相比之下，尽管某些模型在摘要得分上表现良好（如 nomic-embed），但并不一定意味着其事实性或推理能力更强，这表明仅优化长文本语义相似度不足以满足多跳情景下的证据选择需求。除非另有说明，我们在所有实验中均采用 bge-large-en-v1.5。

B.2. 大语言模型主干网络消融实验

我们通过保持检索流水线不变，仅替换 LLM 骨干模型来研究生成器的影响。表 10 展示了各类别结果及总体平均值。总体而言，AtomicRAG 在不同骨干模型上均表现出稳健性：指令微调更强的模型带来了持续的性能提升，但各类别间的相对排名保持稳定，表明主要收益来源于检索质量，而非模型特有的提示技巧。值得注意的是，GPT-4o-mini 达到了最高的平均分，而其他骨干模型则呈现出与模型容量相关的可预测性能下降；然而，即使使用较小的骨干模型（如 Qwen2.5-7B 和 Llama-3.1-8B），该方法仍能维持合理的性能，表明所检索到的原子证据具有足够的针对性，从而减轻了生成器的负担。

Method	Fact	Reason	Summ.	Creat.	Avg.
RAG (w/ rerank)	57.6	56.0	62.0	60.9	59.1
GraphRAG (local)	49.2	61.6	59.0	61.7	57.9
GraphRAG (global)	40.0	61.4	51.1	59.2	52.9
HippoRAG2	64.5	64.1	64.7	60.8	63.5
LightRAG	64.4	64.7	69.4	63.3	65.4
Fast-GraphRAG	62.0	62.4	65.1	63.0	63.1
RAPTOR	50.5	52.9	58.9	61.5	55.9
Ours	68.2	71.1	74.9	68.9	70.8
<i>Improv. vs best baseline</i>	$\uparrow 3.7$	$\uparrow 6.4$	$\uparrow 5.5$	$\uparrow 5.7$	$\uparrow 5.3$

Table 9. 在医学数据集上使用 Qwen2.5-14B 进行生成式评估的准确率 (ACC)。Avg. 为四个类别上的平均值。Improv. vs best baseline 表示我们的方法相对于最佳基准（不包括我们自身）的绝对得分提升（以百分点计）； \uparrow 表示提升。






Vendor	LLM	Fact	Reason	Summ.	Creat.	Avg.	Δ Avg
 OpenAI	GPT-4o-mini	72.6	74.8	<u>76.8</u>	<u>68.3</u>	73.1	—
 DeepSeek	DeepSeek-V3	<u>69.3</u>	<u>73.0</u>	79.5	67.0	<u>72.2</u>	-0.9
 Qwen	Qwen2.5-14B-Instruct	68.2	71.1	74.9	68.9	70.8	-2.3
 Qwen	Qwen2.5-7B-Instruct	64.7	69.2	73.2	65.3	68.1	-5.0
 Meta	Llama-3.1-8B-Instruct	64.7	67.8	70.8	62.5	66.5	-6.6

Table 10. **LLM 骨干网络的消融实验**。我们报告了各类别得分以及总体平均值 (Avg.)。 Δ Avg 表示相对于 GPT-4o-mini 的绝对得分变化。

表 10 显示，**GPT-4o-mini** 的总体得分最高（平均 73.1），表明即使在检索保持不变的情况下，生成器强度依然是一个关键因子。DeepSeek-V3 是最接近的替代方案（72.2，仅落后 0.9 分），并在摘要任务中取得最佳表现（79.5），表明在提供相关证据后，其长文本合成能力更强，尽管在事实和推理导向的任务上略逊于 GPT-4o-mini。Qwen2.5-14B-Instruct 排名第三（70.8），但创造了最高的创意生成得分（68.9），暗示不同模型先验可能更倾向于开放式的风格化补全，而非严格的事实依据。相比之下，两个较小的指令模型得分明显下降（68.1 和 66.5），尤其在复杂推理与创意生成任务上退化最为显著，这与在相同提示和检索上下文下多步证据整合及全局连贯性方面的鲁棒性降低一致。

B.3. 医学领域的额外基准比较

为了补充使用 GPT-4o-mini 作为生成器和评估器的主要结果，我们进一步在统一评估器下，将 AtomicRAG 与具有代表性的 RAG 方法及图增强基线在 **Medical** 分割上进行比较。表 9 报告了在所有方法均采用 Qwen2.5-14B 进行生成式判断时的答案准确率 (ACC)，评估遵循官方 Graph-Bench 协议。该情景固定了评估主干模型，因此性能差异更直接地反映了各系统检索和证据组织流水线的质量，而非不同大语言模型评估器的特殊性。

总体而言，AtomicRAG 的平均得分（70.8）最高，比表现最好的基准方法 LightRAG（65.4）高出超过五个绝对百分点。在所有四个类别中，提升均保持一致：与最强的对比方法相比，AtomicRAG 在事实检索、

Retriever	Fact	Reason	Summ.	Creat.	Avg.	Δ Avg	Time (s/q)
PPR (Baseline)	72.6	74.8	76.8	68.3	73.1	–	<u>0.79</u>
RWR	62.4	70.9	74.5	65.0	68.2	-4.9	0.78
Power Iteration	70.7	<u>73.3</u>	78.6	<u>67.9</u>	<u>72.6</u>	-0.5	18.67
Katz Index	<u>70.7</u>	73.0	<u>76.9</u>	65.7	71.6	-1.5	20.28
Label Propagation	67.2	71.5	71.9	65.5	69.0	-4.1	1.38
Weighted BFS	61.0	68.1	69.8	64.3	65.8	-7.3	13.37

Table 11. 图检索策略的消融实验。我们报告了每个类别得分以及总体平均值 (Avg.)。Time 越低越好。 Δ Avg 表示相对于 PPR 的绝对得分变化。

复杂推理、上下文摘要和创意生成方面均提升了数个百分点，如 *Improv. vs best baseline* 行所总结。这些结果表明，即使在使用强大的外部 LLM 进行标准化评估时，原子级别的证据组织和实体共振检索也能在领域特定的医学问答任务中，提供比基于块或谓词中心图的基线更可靠的证据链。

表 9 列出了带有重排序的 RAG、两种 GraphRAG 检索模式（本地/全局）、HippoRAG2、LightRAG、Fast-GraphRAG、RAPTOR 以及我们的 AtomicRAG。提升幅度 *vs* 基准行报告了 AtomicRAG 相对于每列最强基准的绝对 ACC 提升（以百分点计），在统一评估协议下提供了清晰的性能差距视图。

B.4. 图检索变体

我们比较了在实体-原子图上检索证据原子的代表性图扩散与路径依赖的排序策略。随机游走带重启 (RWR) 通过从查询种子出发的蒙特卡罗可重启游走来近似个性化页面排名 (PPR)。幂迭代通过迭代更新显式求解 PPR 的固定点方程，直至收敛。Katz 指数通过计算从种子到目标结点的路径并按路径长度进行指数衰减来对结点进行排序。标记传播通过对已标记结点在图中进行迭代的标签扩散（平滑）来实现。加权广度优先搜索采用基于跳数的扩展方式，并结合距离衰减权重，生成一种启发式的扩散得分。

表 11 显示，**PPR** 在质量-延迟权衡上表现最佳，以较低的检索延迟（0.79 秒/查询）获得了最高的综合得分（平均 73.1）。RWR 在延迟方面与 PPR 相当（0.78 秒/查询），但准确率大幅下降（平均 68.2， Δ 平均 = -4.9），表明在采样预算下蒙特卡罗估计量产生更高方差且排名更不稳定，这对以证据为中心的任务不利。Power Iteration 和 Katz 在得分上仍具竞争力（平均分别为 72.6 和 71.6），且 Power Iteration 在摘要生成方面甚至有所提升（78.6 对比 76.8），但两者在线推理时速度极慢（18.67 和 20.28 秒/查询），说明每查询收敛和长程聚合主导了运行时间。标记传播相对高效（1.38 秒/查询），但平均表现较差（平均 69.0， Δ 平均 = -4.1），这与过度平滑效应一致，导致相关性信号被稀释。加权 BFS 整体表现最差（平均 65.8），同时速度也较慢（13.37 秒/查询），表明基于跳数的启发式方法既无法可靠逼近稳态分布，又在搜索前沿扩展时难以有效扩展。总体而言，该消融实验支持将 **稳态分布排序 (PPR)** 作为默认检索器：采样近似 (RWR) 牺牲了过多准确率，确切求解器/路径计数 (Power Iteration、Katz) 在计算上与在线检索不匹配，而其他扩散方案（标记传播、加权 BFS）往往模糊了区分性证据。

B.5. PPR 超参数敏感性

我们分析了基于 PPR 的共振检索对两个关键超参数的敏感性：重启/衰减系数 ρ 和原子-种子个性化权重 λ_{seed} 。在所有实验中，我们保持图结构、查询分解、重排序和生成设置不变，仅改变目标超参数。

Damping (ρ)	Fact	Reason	Summ.	Creat.	Avg.
0.1	<u>71.8</u>	72.6	77.2	67.2	<u>72.2</u>
0.2	69.8	72.5	75.7	66.3	71.1
0.3	72.6	74.8	<u>76.8</u>	68.3	73.1
0.4	70.5	<u>72.9</u>	74.9	66.1	71.1
0.5	71.3	<u>73.2</u>	76.5	65.6	71.7
0.6	69.8	72.3	75.1	67.2	71.1
0.7	71.9	72.9	76.5	<u>67.9</u>	72.3
0.8	65.1	70.6	69.6	64.8	67.5
0.9	61.7	67.6	69.2	66.4	66.2

Table 12. 关于重启/衰减系数 ρ 的消融实验。最佳结果用粗体表示，次佳结果用下划线表示。

Atom weight (λ_{seed})	Fact	Reason	Summ.	Creat.	Avg.
0.01	<u>71.1</u>	72.8	<u>76.2</u>	<u>67.5</u>	<u>71.9</u>
0.05	71.3	<u>73.2</u>	76.5	65.6	71.7
0.1	72.6	74.8	<u>76.8</u>	68.3	73.1
0.2	70.0	72.7	75.1	64.4	70.6
0.3	69.6	72.9	74.7	65.2	70.6
0.4	69.9	72.6	74.4	64.4	70.3
0.5	69.2	72.1	73.3	64.0	69.7
0.6	68.4	70.9	73.8	60.4	68.4
0.7	64.2	66.1	69.5	60.9	65.2
0.8	62.7	67.7	69.7	59.4	64.9
0.9	59.5	62.0	64.1	58.7	61.1
1.0	59.0	62.2	62.6	55.3	59.8

Table 13. 消融实验关于原子权重 λ_{seed} 。最佳结果用粗体，次佳结果用下划线。

衰减系数 ρ 。 表 12 清晰地显示出在 $\rho = 0.3$ 附近存在一个“最佳点”，能够获得最优的整体平均得分。较小的取值（例如 $\rho \in [0.1, 0.2]$ ）会强调种子分布周围的局部邻域，这有助于保留摘要生成中的显著证据，但容易导致对更长路径的多跳推理探索不足，从而限制了推理性能的提升。当 ρ 超过 0.5 后，随机游走逐渐被图扩散主导；虽然在某些情况下可略微改善覆盖范围，但也会放大连通性噪声并削弱查询特异性。当 ρ 较大时（例如 $\rho \geq 0.8$ ），所有类别上的性能均出现显著下降，这与过度平滑至高连接度或全局中心实体的现象一致。

原子种子权重 λ_{seed} 。 表 13 调整了个性化向量中分配给原子级种子的质量。我们发现适度的种子设置（ $\lambda_{\text{seed}} = 0.1$ ）最为理想且稳健，表明原子级别的信号应强烈引导传播过程，但仍需为实体级别的扩散留出空间以连接多跳。当 λ_{seed} 过小（例如 0.01）时，PPR 更依赖于粗粒度的实体连接性，降低了选择性并损害事实准确性与创造性。相反，若 λ_{seed} 过大（例如 ≥ 0.3 ），则行走过程变得过于短视：排名被少数紧邻种子的原子所主导，导致跨实体聚合性能下降，影响多跳检索效果，使得性能随 $\lambda_{\text{seed}} \rightarrow 1.0$ 呈单调下降趋势。

总体而言，这些结果表明，AtomicRAG 在一种平衡的机制下受益，其中 PPR 传播既不过于局部也不过于全局，且原子级别的个性化为多跳证据发现提供了强大但非排他性的锚点。除非另有说明，我们使用 $\rho = 0.3$ 和 $\lambda_{\text{seed}} = 0.1$ 。

C. 证明

C.1. 命题 1 的证明：AEG 比基于谓词标注的知识图谱更具全面性和鲁棒性

本节形式化了 AtomicRAG 中使用的原子-实体图 (AEG) 的表示与鲁棒性优势。我们与谓词标注的知识图谱进行对比，这类图谱常用于基于图形的 RAG 中，其中语义内容由提取的谓词类型边承载。

谓词标注的知识图谱基准 谓词标注的知识图谱是一种有向的、带有谓词类型的多重图。

$$G_{\text{KG}} = (\mathcal{E}, \mathcal{R}, \mathcal{T}), \quad \mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E},$$

其中每个三元组 $(h, r, t) \in \mathcal{T}$ 被解释为一个关系断言 $r(h, t)$ 。此基准不包含限定词、重述、来源结点或高阶陈述。

AEG 回顾。 AEG 是一个异构图 $G_{\text{AEG}} = (V, \mathcal{L})$ ，其结点集为 $V = \mathcal{A} \cup \mathcal{E}$ 。语义内容仅由原子 $a \in \mathcal{A}$ 承载，其中每个原子是一个极小的、自包含的命题。图边仅提供组织结构：骨架包含边

$$\mathcal{L}_{\text{cont}} = \{(a, e) \mid a \in \mathcal{A}, e \in \mathcal{E}(a)\}$$

仅编码实体 e 在原子 a 中被提及的结构事实。可选的辅助实体—实体链接作为弱连接线索，不构成谓词类型承诺。

C.1.1. 全面性：将知识图谱嵌入到自适应嵌入图中并保持严格性

Definition C.1 (KG-to-AEG embedding). 给定一个谓词标注的知识图谱 $G_{\text{KG}} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ ，定义一个原子表达图 $\Phi(G_{\text{KG}}) = (V, \mathcal{L})$ 如下。对于每一个三元组 $(h, r, t) \in \mathcal{T}$ ，创建一个原子结点 $a_{h,r,t} \in \mathcal{A}$ ，其原子文本编码命题 $r(h, t)$ ，并设置 $\mathcal{E}(a_{h,r,t}) = \{h, t\}$ 。向 $\mathcal{L}_{\text{cont}}$ 添加包含边 $(a_{h,r,t}, h)$ 和 $(a_{h,r,t}, t)$ 。不向 \mathcal{L} 添加任何谓词类型边。

Definition C.2 (AEG-to-KG projection). 定义一个投影 π_{KG} ，它将 $\Phi(G_{\text{KG}})$ 映射回一个谓词标注的知识图谱，方法是将每个原子 $a_{h,r,t}$ 解析为对应的三元组 (h, r, t) ，并返回所有此类三元组的集合。

Lemma C.3 (AEG can represent any predicate-labeled KG). 对于任意谓词标注的知识图谱 G_{KG} ，我们有 $\pi_{\text{KG}}(\Phi(G_{\text{KG}})) = G_{\text{KG}}$ 。

证明。根据构造，对于每个 $(h, r, t) \in \mathcal{T}$ ， Φ 恰好创建一个编码命题 $r(h, t)$ 的原子 $a_{h,r,t}$ 。投影 π_{KG} 恢复所有此类三元组且仅恢复这些三元组，因此 $\pi_{\text{KG}}(\Phi(G_{\text{KG}})) = (\mathcal{E}, \mathcal{R}, \mathcal{T}) = G_{\text{KG}}$ 。□

引理 C.3 表明，AEG 在表示关系断言方面至少与谓词标注的知识图谱基准一样具有表达能力。

我们现在通过展示 AEG 原生表示的信息来确立严格性（即保持局部语义上下文的独立原子命题），而谓词标注的知识图谱基准则无法表示这些信息，除非将形式化体系扩展至超出谓词类型边的范围。

Definition C.4 (Contextual distinguishability). 一个表示是上下文可区分的，如果它可以表示两个具有相同关系核心（相同 (h, r, t) ）但上下文语义内容不同（例如时间、范围、归属或话语解析的限定词）的证据作为不同的对象，而不会将它们合并。

Lemma C.5 (Predicate-labeled KG is not contextually distinguishable). 在谓词标注的知识图谱基准中，具有相同关系核心 (h, r, t) 的两条证据必然被识别为同一边断言，因此除非模型扩展至超出谓词类型边的范围，否则它们的上下文差异将丢失。

证明。在基准知识图谱中，语义载体是带类型的边 $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ 。如果两个证据共享相同的 (h, r, t) ，它们会映射到 \mathcal{T} 的同一元素。基准结构没有额外的组件来编码不同的上下文，同时保持它们的区分性，除非引入额外的对象（例如，重新表述/限定词/来源结点），而这超出了基准的定义范围。因此，基准不具备上下文可区分性。□

Theorem C.6 (AEG is strictly more comprehensive than predicate-labeled KG). 存在一个上下文可区分的 AEG, 而没有任何谓词标注的知识图谱基准能够表示相同的信息, 除非将形式化扩展至超出谓词类型边的范围。

证明. 考虑两个原子 $a_1, a_2 \in \mathcal{A}$, 它们共享相同的关联核心 (h, r, t) , 但在上下文语义上存在差异: a_1 断言在上下文 c_1 下 $r(h, t)$ 成立, 而 a_2 断言在另一个上下文 c_2 下 $r(h, t)$ 成立, 且 $c_1 \neq c_2$ (其中上下文可编码时间窗口、作用域、归属关系或任何话语解析后的限定条件)。根据 AtomicRAG 中的原子定义, 每个 a_i 均为一个极小的自包含命题, 可作为独立的语义对象存储于 AEG 中。这两个原子均通过包含边与实体相连, 因此均可被检索并组合使用。

相比之下, 谓词标注的 KG 基准必须将两种证据合并为同一条类型边 (h, r, t) , 从而丢失了 c_1 与 c_2 之间的区别, 根据引理 C.5。因此, 该 AEG 所表示的信息无法在基准 KG 中表示, 除非扩展形式化体系。这证明了严格全面性。 \square

C.1.2. 鲁棒性: 解耦语义可减少由噪声谓词边引起的传播泄漏

传播模型。 令 P 表示个性化 PageRank (PPR) 中使用的行规范化转移矩阵。给定一个个性化向量 π , PPR 满足

$$\mathbf{r} = \rho \pi + (1 - \rho) P^\top \mathbf{r}, \quad \rho \in (0, 1).$$

将结点划分为相关区域 R 和无关区域 I (取决于查询)。考虑一个两区域宏观状态转移矩阵

$$T = \begin{pmatrix} 1 - \gamma & \gamma \\ \varepsilon & 1 - \varepsilon \end{pmatrix}, \quad e = (1, 0),$$

其中 γ 是在随机游走下通过跨区域边从 R 离开到 I 的概率, 而 ε 是从 I 回到 R 的总回报概率。

Lemma C.7 (Relevant mass under two-region PPR and monotone leakage). 令 $\varphi = (\varphi_R, \varphi_I)$ 为在宏链上由 PPR 诱导出的关于 $\{R, I\}$ 的稳态分布。

$$\varphi_R = \frac{\rho + (1 - \rho)\varepsilon}{\rho + (1 - \rho)(\gamma + \varepsilon)}.$$

特别地, 如果 $\varepsilon \approx 0$, 那么

$$\varphi_R \approx \frac{\rho}{\rho + (1 - \rho)\gamma},$$

且 φ_R 关于 γ 严格递减。

证明. 宏 PPR 不动点方程为 $\varphi = \rho e + (1 - \rho)\varphi T$, 其中 $\varphi_R + \varphi_I = 1$ 。求解第一个坐标可得闭式解。对 γ 求导得到严格负的导数, 因此 φ_R 随 γ 严格递减。 \square

引理 C.7 表明, 基于传播的检索的鲁棒性可归结为对跨区域泄漏参数 γ 的控制。

Assumption C.8 (Predicate-edge noise induces larger leakage than containment backbone). 对于固定的语料库和抽取流水线, 虚假的谓词类型边引入跨区域转移的频率至少与包含边一样频繁。此外, 在 AEG 中, 辅助实体-实体边 (包括关联或同义链接) 被降权, 使得其总的转移概率贡献相对于主干包含转移被限制在一个因子 $\beta \in (0, 1)$ 之内。

Theorem C.9 (AEG yields smaller propagation leakage than predicate-labeled KG). 在假设 C.8下，由 AEG 引起的有效跨区域泄漏参数满足 $\gamma_{\text{AEG}} \leq \gamma_{\text{KG}}$ 对于谓词标注的知识图谱基准。因此，相关的平稳质量满足 $\varphi_R^{\text{AEG}} \geq \varphi_R^{\text{KG}}$ ，当泄漏不同时为严格不等式。

证明. 在谓词标注的知识图谱基准中，实体结点之间的转移通过谓词类型的边直接实现。抽取错误可能会产生虚假的跨区域边，这些边会完全贡献于离开 R 的随机游走概率，从而增加 γ_{KG} 。

在 AEG 中，主干转移由包含边介导：从实体 e 到原子 a 的转移需要 $e \in \mathcal{E}(a)$ ，从 a 到另一个实体 e' 的转移需要 $e' \in \mathcal{E}(a)$ 。这些转移在结构上受到提及局部性的约束，并根据假设 C.8，其引发跨区域跳跃的倾向性不大于虚假谓词边。辅助的实体-实体边被降权，使其贡献被限制在 β 以内。因此，在 AEG 中离开 R 的总概率质量不大于谓词标注知识图谱基准中的情况，这意味着 $\gamma_{\text{AEG}} \leq \gamma_{\text{KG}}$ 。

最后，引理 C.7 表明 φ_R 随 γ 单调递减，因此 $\varphi_R^{\text{AEG}} \geq \varphi_R^{\text{KG}}$ ，当 $\gamma_{\text{AEG}} < \gamma_{\text{KG}}$ 时为严格不等式。 \square

结合定理 C.6 (严格完备性) 和定理 C.9 (传播鲁棒性) 可得命题 1。

C.2. 命题 2 的证明：粒度对齐有助于检索

本节证明了双向粒度不匹配原理，并说明了 AtomicRAG 的原子级存储和查询分解为何能将检索过程带入有利状态。

C.2.1. 统一的形式化：将检索视为对证据集的排序

令 \mathcal{A} 表示原子证据项 (AtomicRAG 中的原子) 的宇宙。对于一个查询 q ，假设存在一个最小充分证据集 $A^*(q) \subseteq \mathcal{A}$ ，该集合是支持正确答案所必需的，并记为 $m := |A^*(q)|$ 。一个检索单元 U (例如，块、子图、社区、路径、三元组或原子) 可序列化为一个证据集 $C(U) \subseteq \mathcal{A}$ 。定义

$$r(U) := |A^*(q) \cap C(U)|, \quad M(U) := |C(U)|.$$

定义覆盖度和纯度：

$$\text{Cov}(q, U) = \frac{r(U)}{m}, \quad \text{Pur}(q, U) = \frac{r(U)}{M(U)}.$$

C.2.2. 粗粒度单元稀释：分离度随纯度变化，集合大小增大时误排序情况加剧

假设一个原子级别的评分模型 $s(q, a)$ ：

$$s(q, a) = \mu_{Y(a)} + \varepsilon_a, \quad Y(a) = \mathbf{1}[a \in A^*(q)],$$

其中 $\mu_1 > \mu_0$ 、 $\Delta\mu = \mu_1 - \mu_0 > 0$ 和 ε_a 为独立同分布的 σ 次高斯随机变量。对单元进行秩排序时，采用平均聚合：

$$S(q, U) = \frac{1}{M(U)} \sum_{a \in C(U)} s(q, a).$$

Lemma C.10 (Expected score gap equals purity-scaled signal). 令 U^+ 有 $r := r(U^+) \geq 1$ 和 $M := M(U^+)$ ，且令 U^- 满足 $r(U^-) = 0$ 和 $M(U^-) = M$ 。则

$$\mathbb{E}[S(q, U^+)] - \mathbb{E}[S(q, U^-)] = \frac{r}{M} \Delta\mu.$$

证明. 展开 $S(q, U)$ 并取期望值会消除均值为零的噪声项, 仅留下必要原子均值之差, 该差值被单位中包含的必要原子比例 r/M 缩放. \square

Theorem C.11 (Misranking probability bound degrades with coarse evidence sets). 在与引理 C.10 相同的设定下,

$$\mathbb{P}(S(q, U^-) \geq S(q, U^+)) \leq \exp\left(-\frac{r^2(\Delta\mu)^2}{4\sigma^2 M}\right).$$

证明. 令 $D = S(q, U^+) - S(q, U^-) = \frac{r}{M}\Delta\mu + X$, 其中 X 为两个独立的次高斯噪声平均值之差. 根据次高斯封闭性, X 是 $(\sqrt{2}\sigma/\sqrt{M})$ -次高斯的. 标准尾部界可得

$$\mathbb{P}(D \leq 0) = \mathbb{P}\left(X \leq -\frac{r}{M}\Delta\mu\right) \leq \exp\left(-\frac{r^2(\Delta\mu)^2}{4\sigma^2 M}\right).$$

\square

定理 C.11 表明, 粗粒度的检索单元 (大 M 且小 r) 会导致分离能力弱和排名不稳定.

C.2.3. 细单元在顶层- k 下因覆盖范围限制而失效

Theorem C.12 (Coverage upper bound for overly fine units under top- k). 假设对于每个检索单元 U 均满足 $|C(U)| \leq c$ (例如, 针对三级单元的 $c = 1$). 对于任意 top- k 集合 $\{U_1, \dots, U_k\}$,

$$\left|A^*(q) \cap \bigcup_{i=1}^k C(U_i)\right| \leq kc, \quad \text{Cov}(q, \{U_i\}_{i=1}^k) \leq \frac{kc}{m}.$$

特别地, 如果 $kc < m$, 则无论排名质量如何, 完全覆盖都是不可能的.

证明. 由于 $A^*(q) \cap \bigcup_{i=1}^k C(U_i) \subseteq \bigcup_{i=1}^k C(U_i)$,

$$\left|A^*(q) \cap \bigcup_{i=1}^k C(U_i)\right| \leq \left|\bigcup_{i=1}^k C(U_i)\right| \leq \sum_{i=1}^k |C(U_i)| \leq kc.$$

除以 m 得到覆盖的界. \square

C.2.4. 为什么原子 RAG 能改进系统: 原子级单元加上查询分解

AtomicRAG 通过以下方式实现粒度对齐: (i) 使用自包含的原子单元作为语义载体, 避免粗粒度单元导致的信息稀释; (ii) 将复杂查询分解为少量原子子查询, 降低每次检索实例的有效证据需求.

令有效查询集为 $\tilde{\mathcal{Q}}(q)$, 如正文所述. 为了保证 LaTeX 的鲁棒性, 我们引入一个简写:

$$\tilde{\mathcal{Q}}_q := \widetilde{\mathcal{Q}(q)}.$$

对于每个 $q' \in \tilde{\mathcal{Q}}_q$, 令 $A^*(q')$ 表示大小为 $m_{q'} = |A^*(q')|$ 的最小充分证据集. 定义总体目标证据为

$$A_{\text{all}}^*(q) = \bigcup_{q' \in \tilde{\mathcal{Q}}_q} A^*(q').$$

Corollary C.13 (Decomposition relaxes top- k coverage constraints). 假设对每个 $q' \in \tilde{\mathcal{Q}}_q$ 独立进行检索，且具有相同的 top- k 预算和单元大小约束 $|C(U)| \leq c$ 。那么每个子查询实现完全覆盖仅需 $kc \geq m_{q'}$ ，而非 $kc \geq |A_{\text{all}}^*(q)|$ 。因此，当 $\max_{q' \in \tilde{\mathcal{Q}}_q} m_{q'} \ll |A_{\text{all}}^*(q)|$ 时，分解扩大了完全覆盖检索的可行区域。

证明. 将定理 C.12 分别应用于每个子查询 q' 。完全覆盖的必要条件变为对每个 q' 的 $kc \geq m_{q'}$ 。如果分解降低了每个子查询的最大证据需求，则相应地放松约束。□

最后，由于原子被定义为极小的自包含命题，AtomicRAG 可以在保持相关单元之间非平凡重叠 $r(U)$ 的同时，使每个语义单元的 $M(U)$ 保持较小。这提高了纯度 $r(U)/M(U)$ ，并增强了定理 C.11 中的误排序指数。这确立了命题 2。

D. 案例研究

图 6 展示了一个复合用户查询，该查询在多个显著实体（如皮肤白皙、日光浴床、基底细胞癌（BCC）、治疗选项）的情景下，同时要求因果解释（“为什么”）和可操作建议（“怎么做”），并隐含了一个多跳推理链（风险因子 \rightarrow 机制 \rightarrow 治疗）。在标准的基于分块的 RAG 流水线中，单次检索往往将病因学和治疗学的异质证据聚集到一个上下文窗口中，这加剧了冗余性，并引入主题偏移（例如，患病率统计量或解剖分布），最终削弱了因果基础，并模糊了问题中以治疗为中心的部分。

AtomicRAG 通过显式分离知识索引（哪些证据属于哪个子意图）与知识表征（证据单元如何编码和去重）来解决这一失效模式。系统首先估计查询复杂度；在此例中，得分超过分解阈值（ $7.0 > 6.5$ ），触发原子问题分解。查询被拆分为两个具有不同关注点的原子子查询： Q_1 关注晒床使用、白皙皮肤与基底细胞癌风险之间的关系； Q_2 关注基底细胞癌的以实体为中心的治疗方案。这种分解从结构上避免了跨领域干扰：病因学证据被检索并整合用于 Q_1 ，而治疗学证据被检索并整合用于 Q_2 。

对于每个原子查询，实体共振图检索执行命名实体识别以获取实体集合，然后在实体-原子图上运行基于图形的传播（例如，个性化 PageRank），以揭示候选知识原子。与块检索相比，原子级别的证据单元在下游聚合过程中提升了可控性，因为它们不仅更小（减少了不可约噪声），而且由实体显式索引（支持组合式证据追踪）。然而，图检索仍可能揭示近似重复项和弱相关原子。因此，AtomicRAG 引入了一个原子筛，其作用为：(i) 消除重复的原子（例如，多个改写版本均指出浅肤色个体使用室内日光浴会增加皮肤癌风险），以及 (ii) 过滤掉与当前子查询无关的原子（例如，发病率数据、常见解剖部位，或关于其他类型皮肤癌的陈述）。经过筛选后， Q_1 保留的原子集中于从日光浴床紫外线暴露到基底细胞 DNA 损伤，以及浅肤色人群易感性增强之间的因果路径；而 Q_2 保留的原子则聚焦于治疗决策（手术作为常见的一线选择，放疗或全身治疗则根据病例因素决定）以及早期检测/筛查在规划中的作用。

最后，生成器通过融合两个精心筛选的原子集合来组成响应，从而产生一个清晰分离因果解释与治疗建议的答案，同时保持明确的证据链。此示例突显了 AtomicRAG 在实践中的核心优势：通过将证据表示为原子事实，并围绕分解后的意图组织检索，该系统减少了冗余，限制了领域泄露，并在复合查询目标下稳定了多跳推理。

E. 提示模板

AtomicRAG 依赖一组少量的提示模板，用于实现 (i) 语料库到结构的构建，(ii) 查询分解与证据选择，以及 (iii) 答案生成与评估。为保持附录的可读性，此处仅简要总结每个提示族的作用，并以图表形式提供完整的模板内容，以确保确切的可复现性（见图 7–12）。

命名实体识别 我们使用实体抽取提示来识别每个段落中的关键命名实体。输出是一个结构化的 JSON 列表，随后被下游的抽取提示重用，以鼓励生成基于实体的三元组和片段（图 7）。

统一的三元组与知识原子抽取。 该提示同时从同一段落中提取 (a) RDF 风格三元组和 (b) 自包含知识原子，并施加显式约束以解决指代消解、保留数量/时间跨度并避免冗余片段。它返回一个包含三元组、原子及原子级别实体提及的单一 JSON 对象（图 8）。

原子问题分解。 我们采用单次调用的分解提示，首先对问题复杂度进行评分，然后（仅在需要时）生成一组带有聚焦标签的原子子问题。这确保了分解仅在必要时使用，并且保持可检索-可操作性（图 9）。

知识原子过滤 给定用户问题和候选知识原子集，过滤提示仅选择直接有助于回答问题的原子 ID。输出被限制为一个索引列表的 JSON 格式，防止模型虚构新的证据（图 10）。

摘要问答与精确问答。 我们使用两种阅读理解提示来进行答案综合：(i) 一个抽象问答提示，生成包含简要证据推理的完整且自包含的答案；(ii) 一个精确问答提示，在基准要求短格式回答时输出简洁的最终答案（图 11 和 12）。

F. 局限性

AtomicRAG 的一个局限性在于其有效性依赖于离线原子-实体图构建和在线查询分解的质量与一致性。特别是，原子化步骤和实体正则化通常由指令微调的 LLM（或自动化流水线）生成，因此抽取质量的差异会渗透到下游的图连接性和检索邻域中。同样，当启用原子化问题分解时，分解粒度和子问题覆盖范围会影响后续传播和筛选步骤识别正确证据链的能力。尽管我们的设计减少了对谓词标注的依赖，并在噪声环境下具有较强的鲁棒性，但对于需要严格关系语义的任务（例如细粒度的时间/因果约束），其表达能力可能不足，此时额外的关系感知信号可进一步提升性能。最后，AtomicRAG 引入了额外的系统组件（图存储、索引及可选的分解模块），虽然我们报告了效率结果，但要实现对持续更新语料库（频繁插入/删除）的缩放，可能还需要额外的工程工作以支持增量更新和稳定性。

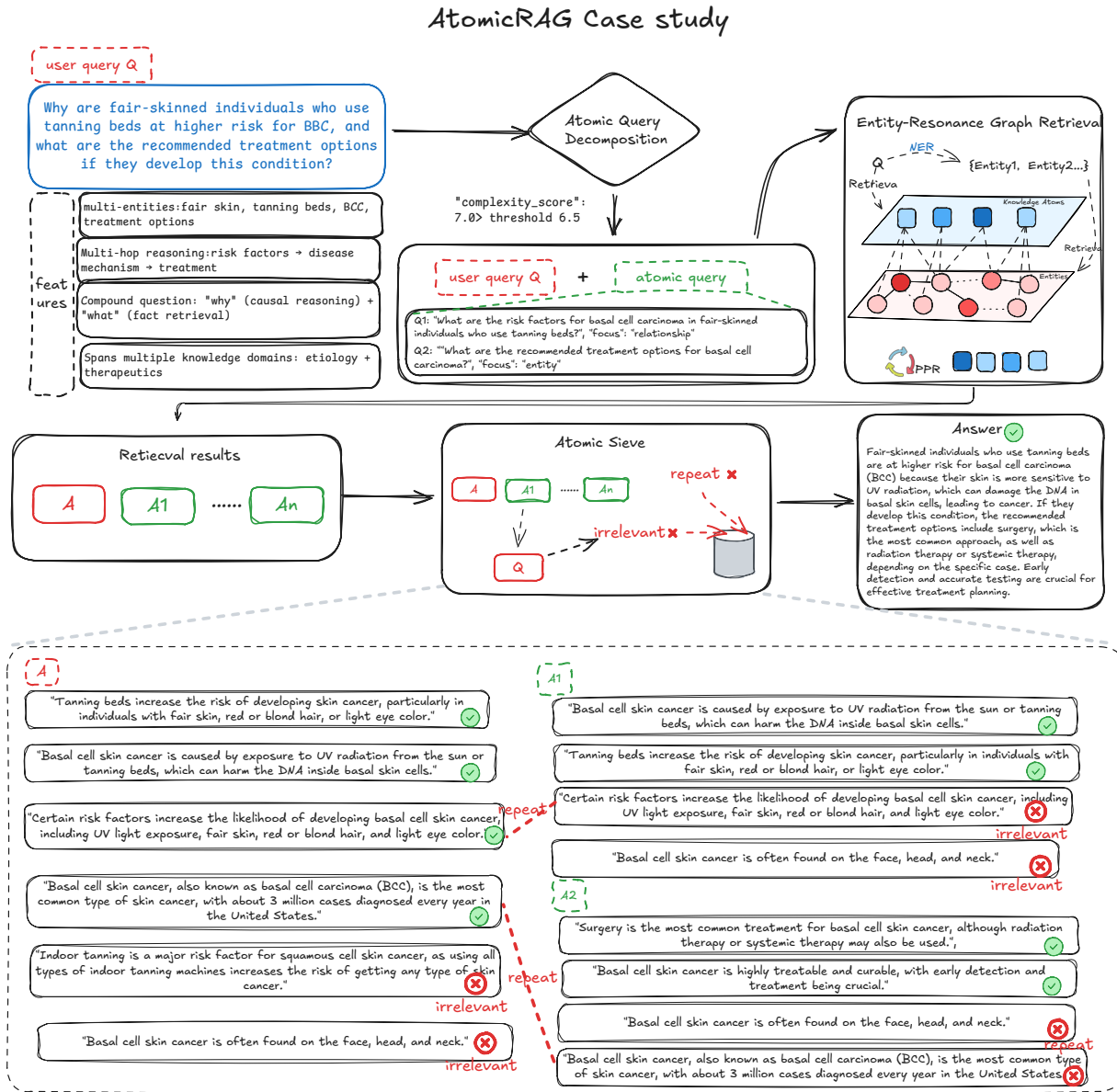


Figure 6. 案例研究。

```

System Prompt:
'''
Your task is to extract named entities from the given paragraph.
Respond with a JSON list of entities.
'''

One-shot Demonstration:
User Input:
'''
Radio City
Radio City is India's first private FM radio station and was started on 3 July 2001.
It plays Hindi, English and regional songs.
Radio City recently forayed into New Media in May 2008 with the launch of a music
portal - PlanetRadiocity.com that offers music related news, videos, songs, and
other music-related features.
'''

Assistant Output:
```json
{"named_entities":
 ["Radio City", "India", "3 July 2001", "Hindi", "English", "May 2008",
 "PlanetRadiocity.com"]}
}
'''

```

Figure 7. 命名实体识别 (NER) 的提示模板。

```

System Prompt:
'''
Your task is to extract both RDF triples and knowledge fragments from the given passage in a single unified process.

Requirements for RDF triples:
- Each triple should contain at least one, but preferably two, of the named entities from the provided list - Follow the format: [subject, predicate, object] - Clearly resolve pronouns to their specific names to maintain clarity - Focus on factual relationships present in the text
Requirements for knowledge fragments:
- Each fragment must be self-contained with clear boundaries - Focus on concrete facts, discoveries, achievements, or specific processes- Include essential context (who, what, when, where) within the fragment - Preserve precise numerical values, time periods, ranges, and measurements - Include complete causal chains and deeper applications/implications - Prefer specific, verifiable information over general statements - Fragments should be atomic - covering one complete idea or fact - Avoid redundant or overlapping information between fragments - Each fragment should be retrievable and useful on its own
Prioritize knowledge fragments that contain:
- Scientific discoveries and their implications with clear temporal context - Historical developments with clear outcomes and time periods - Technical processes and their specific applications - Causal relationships with measurable effects - Quantifiable facts and measurements - Complete purpose/application chains (not just what, but why and for what purpose)
For each knowledge fragment, identify which entities from the triples are mentioned in it to ensure consistency.
Respond with a JSON object containing "triples", "knowledge_fragments", and "fragment_entities" lists.
'''

```

Figure 8. 统一三元组与知识原子抽取的提示模板。

**System Prompt:**

```
```You are an expert assistant for a knowledge-graph RAG system.
Decide if the user question needs decomposition and, if yes, write atomic
sub-questions that cover the whole ask. Question: ${question}
Decompose when:- Multiple entities/relations or multi-hop reasoning is needed - "How/why/compare" reasoning that
spans steps - Compound prompts unlikely to be answered by one fact
Do NOT decompose when: - One fact answers it (who/what/when/where) - Only one entity/relation with a direct
answer
If you decompose, write up to ${max_sub_questions} sub-questions. Each must:1) Cover exactly one aspect 2) Be
answerable independently via retrieval 3) Stay in the same domain/context
Focus label options: "entity", "relationship", "reasoning", "context".
Return JSON only:
{
  "needs_decomposition": true/false,
  "complexity_score": number 0-10,
  "reasoning": "short justification",
  "sub_questions": [
    {"id": 1, "question": "...", "focus": "entity"}
  ]
}
- If needs_decomposition is false, sub_questions = [].
Output:
```
```

Figure 9. 问题复杂度评分与原子分解提示模板



**System Prompt:**

'''

Your input fields are:

1. `question` (str): The user's question2. `fragments\_before\_filter` (str): Knowledge fragments to be filtered

Your output fields are:

1. `kept\_fragment\_ids` (KeptFragmentIndices): JSON object describing which fragment IDs to keep

All interactions will be structured as:

[[ question ]]{question}[[ fragments\_before\_filter ]]{fragments\_before\_filter}[[ kept\_fragment\_ids ]]{kept\_fragment\_ids} # JSON: {"keep\_indices": [0, 2, ...]}

[[ completed ]]Objective: Filter knowledge fragments to keep ONLY those directly relevant to answering the question.

Each fragment includes an `id` and the `text`. Choose the ids that should be kept.

KEEP fragments that: - Directly answer the question - Provide essential context for understanding the answer -

Contain key facts needed for reasoning

REMOVE fragments that:

- Discuss tangential topics
- Mention similar but different entities
- Provide generic background not needed for this specific question

Return kept fragment ids in JSON format: {"keep\_indices": [id1, id2, ...]}

Return an empty array if none are relevant: {"keep\_indices": []}

Do NOT invent new ids. Only choose from the provided fragment ids.

''''''

Figure 10. 知识单元过滤提示模板

**System Prompt:**

'''

As an advanced reading comprehension assistant, analyze the provided passages and questions meticulously. Begin your response after "Thought:" with a step-by-step explanation that shows how you synthesize the evidence. Conclude with "Answer:" followed by a complete, self-contained response that addresses the question with all essential details.'''

**One-shot Demonstration:**

\*Context Documents:\*

'''

Wikipedia Title: Southampton

The University of Southampton, which was founded in 1862 and received its Royal Charter as a university in 1952, has over 22,000 students. The university is ranked in the top 100 research universities in the world...

Wikipedia Title: Neville A. Stanton

Neville A. Stanton is a British Professor of Human Factors and Ergonomics at the University of Southampton. Prof Stanton is a Chartered Engineer (C.Eng), Chartered Psychologist (C.Psychol) and Chartered Ergonomist (C.ErgHF)...

'''

\*Question:\* When was Neville A. Stanton's employer founded?

\*Expected Output:\*

'''

Thought: The employer of Neville A. Stanton is the University of Southampton, and historical records state the university was founded in 1862.

Answer: The University of Southampton was founded in 1862.

'''

Figure 11. 抽象问答的提示模板。

**System Prompt:**

'''

As an advanced reading comprehension assistant, your task is to analyze text passages and corresponding questions meticulously. Your response start after "Thought: ", where you will methodically break down the reasoning process, illustrating how you arrive at conclusions. Conclude with "Answer: " to present a concise, definitive response, devoid of additional elaborations.'''

**One-shot Demonstration:**

\*Question:\* When was Neville A. Stanton's employer founded?

\*Expected Output:\*

'''

Thought: The employer of Neville A. Stanton is University of Southampton. The University of Southampton was founded in 1862.

Answer: 1862.

'''

Figure 12. 用于精确问答的提示模板。