

BioQA: Retrieval-Augmented Question Answering for Biomedical Literature

Yasyf Mohamedali^{1*} and Matt Thomson²

^{1*}Aneta.

²Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA, USA.

*Corresponding author(s). E-mail(s): yasyf@aneta.company;

Abstract

Answering complex biomedical questions requires synthesizing evidence across thousands of papers, reconciling conflicting findings, and grounding every claim in verifiable sources. We present BioQA, a retrieval-augmented question-answering system that addresses these challenges through four architectural innovations: (i) a hierarchical research tree that decomposes complex queries into sub-question DAGs managed by specialized reasoning agents for query planning, sub-question generation, and adversarial challenge; (ii) a hybrid multi-source retrieval pipeline combining dense and sparse retrieval, knowledge graph community traversal, hypothetical document embeddings, and live web search with adaptive follow-up retrieval; (iii) a multi-signal confidence estimation framework that triangulates self-reflection, observed consistency, and formal verification via an SMT solver; and (iv) query-adaptive output generation that classifies queries into factoid answers, structured tables, literature reviews, and detailed reports, routing each to a specialized generation pipeline. BioQA ingests literature from preprint servers, academic metadata repositories, and patent archives through a distributed pipeline that extracts biomedical entities, resolves mentions against multiple ontologies, and constructs a knowledge graph with hierarchical community structure. We evaluate BioQA using sixteen metrics spanning four quality dimensions—answer correctness, retrieval effectiveness, citation accuracy, and confidence calibration—and compare against peer systems on a curated dataset of biomedical questions. Experimental results demonstrate improvements across retrieval precision, answer quality, and citation accuracy dimensions. An expert evaluation study with domain specialists assesses factual accuracy, evidence grounding, and comprehensiveness through blinded pairwise comparison.

Keywords: biomedical question answering, retrieval-augmented generation, knowledge graphs, multi-agent reasoning

1 Introduction

Biomedical knowledge is accumulating faster than any individual or team can absorb it. PubMed now indexes over 36 million citations, with more than 1.5 million added each year, and the pace is accelerating across preprint servers, patent filings, and clinical trial registries. A clinician evaluating a treatment decision, a drug-discovery scientist assessing a novel target, or a policy-maker reviewing pandemic preparedness must synthesize evidence scattered across hundreds of papers—often spanning molecular biology, epidemiology, and clinical medicine simultaneously. The question-answering benchmarks designed to measure progress on this task expose a persistent gap between machine and human performance. On PubMedQA [1], the best automated system achieves 68.1% accuracy against a 78% human ceiling. BioASQ [2] reveals that even strong models struggle with multi-document summarization, where answers must reconcile conflicting evidence from heterogeneous sources. MedQA [3], which requires multi-step clinical reasoning grounded in medical knowledge rather than passage retrieval alone, further underscores the distance between pattern-matching and genuine understanding.

Large language models (LLMs) have narrowed parts of this gap. GPT-4 exceeds the United States Medical Licensing Examination passing threshold by over 20 percentage points [4], and Med-PaLM 2 achieves 86.5% on MedQA [5], approaching expert-level performance on closed-book medical reasoning. Yet these gains come with a critical liability: without grounding in source literature, LLMs fabricate citations and invent findings with striking confidence. An evaluation by the OpenScholar team found that GPT-4o hallucinated between 78% and 90% of its cited references when answering scientific questions [6]. Meta’s Galactica [7], trained specifically on scientific text, was withdrawn within three days of public release after generating authoritative-sounding but factually incorrect content. In biomedicine, where a hallucinated drug interaction or fabricated clinical trial result could propagate into patient care, ungrounded generation is not merely an inconvenience—it is a safety hazard.

Retrieval-augmented generation (RAG) [8] addresses the grounding problem by conditioning LLM outputs on retrieved evidence. Recent systems have demonstrated that retrieval can substantially improve both accuracy and verifiability. PaperQA2 [9] achieves 85.2% citation precision while parsing an average of 14.5 papers per question. OpenScholar [6], an 8-billion-parameter model with iterative retrieval and self-feedback, outperforms GPT-4o by 5% on correctness metrics for scientific literature synthesis. Self-RAG [10] introduces reflection tokens that allow a model to decide when and what to retrieve, improving factual accuracy on knowledge-intensive tasks. However, these systems share a fundamental architectural limitation: they operate as monolithic single-pass or iterative-feedback pipelines that retrieve, then generate, without structured multi-step reasoning over the retrieved evidence. When a question requires decomposing a complex query into sub-questions, synthesizing evidence across disparate sub-fields, or adapting the output format to the nature of the query, flat retrieve-then-generate pipelines reach their ceiling.

No existing system combines multi-agent orchestrated reasoning with knowledge graph integration, calibrated multi-signal confidence estimation, and query-adaptive output generation within a unified architecture. We present BioQA, an end-to-end

biomedical question-answering system designed to address these limitations through five architectural innovations. First, BioQA decomposes complex queries into a *hierarchical research tree*—a directed acyclic graph of sub-questions that mirrors the structure of a systematic literature review, managed by specialized agents for query planning, sub-question generation, claims extraction, and adversarial challenge, enabling multi-hop reasoning across nested topics (Section 2.3). Second, a *hybrid multi-source retrieval pipeline* combines dense and sparse retrieval, knowledge graph traversal [11], and live web search in a fan-out/rerank architecture with adaptive follow-up retrieval [12] (Section 2.2). Third, a *multi-signal confidence estimation* framework triangulates self-reflective assessment [13], observed consistency across independent answer generations [14], and formal verification of quantitative claims via an SMT solver [15] (Section 2.4). Fourth, a *query-adaptive output generation* layer classifies queries into multiple output types—factoid answers, structured tables, literature reviews, and detailed reports—routing each to a specialized generation pipeline tailored to the expected output structure. Fifth, a *comprehensive evaluation framework* with sixteen metrics spanning four quality dimensions enables systematic assessment of answer correctness, retrieval effectiveness, citation accuracy, and confidence calibration (Section 3). Unlike OpenScholar [6], which achieves strong results through iterative self-feedback over a flat retrieval corpus, BioQA couples structured knowledge graph reasoning with hierarchical query decomposition. The two approaches are complementary: BioQA’s multi-agent design targets complex, multi-hop questions that benefit from explicit research planning, while self-feedback excels at refining single-pass answers through iterative citation verification.

Our main contributions are as follows:

- We introduce a hierarchical research tree architecture with multi-agent reasoning for biomedical question answering that decomposes complex queries into structured sub-question DAGs, with dynamic branch pruning, merging, and reparenting driven by intermediate findings, managed by cooperating agents for planning, analysis, and synthesis.
- We present a hybrid multi-source retrieval pipeline that fuses dense, sparse, knowledge graph, and web retrieval through reciprocal rank fusion and two-stage reranking, with adaptive follow-up queries when evidence gaps are detected.
- We propose a multi-signal confidence estimation framework combining self-reflection, observed consistency via NLI-based entailment, and formal verification via constraint satisfaction, providing calibrated uncertainty estimates for biomedical answers.
- We design a comprehensive evaluation framework with sixteen metrics spanning answer quality, retrieval effectiveness, citation accuracy, and calibration, enabling systematic comparison across system configurations and against peer systems.
- We evaluate BioQA through automated metrics and expert evaluation, demonstrating improvements across retrieval, answer quality, and citation accuracy dimensions with blinded pairwise comparison by domain specialists.

The remainder of this paper is organized as follows. Section 2 describes the BioQA system architecture, covering data ingestion and knowledge store construction

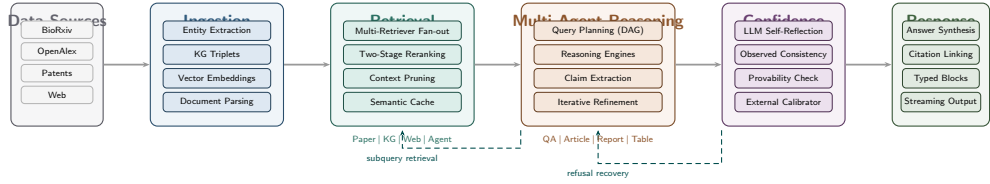


Fig. 1: BioQA system architecture. Queries flow left to right through six stages. *Data Sources* are ingested into vector stores and a biomedical knowledge graph. A multi-retriever fan-out with learned reranking feeds a multi-agent reasoning layer that decomposes queries into a research tree (DAG), dispatches specialized engines, and iteratively refines answers. Confidence scoring aggregates four independent signals before the final response is synthesized with inline citations. Dashed arrows indicate iterative feedback loops for subquery retrieval and refusal recovery.

(Section 2.1), the multi-source retrieval pipeline (Section 2.2), multi-agent reasoning and answer synthesis (Section 2.3), confidence estimation (Section 2.4), and deployment infrastructure (Section 2.5). Section 3 introduces our evaluation methodology and metrics. Section 4 presents experimental results, including comparisons against baseline systems and ablation studies isolating the contribution of each architectural component. Section 5 reports expert evaluation by domain specialists. Section 6 situates BioQA within the broader landscape of biomedical question answering, retrieval-augmented generation, and multi-agent reasoning. Section 7 discusses limitations and directions for future work.

2 BioQA System

BioQA is an end-to-end biomedical question-answering system that integrates multi-source literature ingestion, knowledge graph construction, multi-stage retrieval, hierarchical multi-agent reasoning, and calibrated confidence estimation into a unified pipeline. A user query first passes through a disambiguation stage that resolves ambiguous biomedical terminology and classifies the query into one of multiple output types—ranging from factoid answers and structured tables to full literature reviews and rendered reports. The classified query is routed to a specialized reasoning engine (Section 2.3), which decomposes it into a research tree of sub-questions, each answered via a retrieval-augmented generation (RAG) pipeline (Section 2.2) operating over a knowledge store built from millions of biomedical papers, preprints, and patents (Section 2.1). Sub-question answers are synthesized into a final response, accompanied by a multi-signal confidence score (Section 2.4) and grounded inline citations. Unlike systems that rely on iterative self-feedback over a flat retrieval corpus [6], BioQA’s architecture couples a structured knowledge graph with hierarchical query decomposition, enabling it to answer complex multi-hop questions that require integrating evidence across disparate subfields.

2.1 Data Ingestion and Knowledge Store

The knowledge store underpinning BioQA is built by a distributed ingestion pipeline that continuously processes biomedical literature from four primary source categories: preprint servers (bioRxiv, medRxiv, with coverage from 2013 to the present), academic metadata repositories (OpenAlex, Semantic Scholar, filtered to core biomedical and life-science fields), patent archives (USPTO, with structured claims extraction), and curated clinical sources. An on-demand pathway additionally ingests individual papers by DOI, enabling rapid incorporation of newly published work without waiting for a full crawl cycle. Raw documents are fetched by source-specific crawlers, then parsed through a strategy-based PDF extraction layer that cascades through multiple backends with graceful degradation—if the primary parser fails on a document, the pipeline automatically attempts alternative extraction strategies, ensuring robustness across diverse PDF layouts including multi-column formats, scanned manuscripts, and supplementary materials.

Parsed text undergoes a two-stage chunking procedure. A neural chunking model based on a large pre-trained language representation architecture first segments documents into semantically coherent passages, after which a science-domain sentence embedding model refines boundaries by detecting semantic shifts between adjacent sentences. The resulting chunks are organized into a hierarchical node tree that preserves document structure—sections, subsections, paragraphs, and captions—as parent-child relationships, enabling retrieval at multiple granularity levels from individual passages to entire sections.

A three-stage data point extraction pipeline operates over each chunk, targeting seven section types (Abstract, Results, Discussion, Claims, Limitations, Conclusion, and Summary). In the first stage, an LLM identifies candidate data points—quantitative findings, experimental outcomes, and key assertions. In the second stage, each candidate is classified by confidence level: exact (directly stated numerical values), approximate (ranges or qualified estimates), estimated (inferred from context), or reported (attributed to external sources). In the third stage, confidence levels are mapped to relevance scores that weight the data point’s contribution during downstream retrieval and synthesis. This structured extraction preserves the evidential strength of source material, enabling the retrieval pipeline to prioritize high-confidence findings.

The entire ingestion pipeline is distributed across hundreds of concurrent workers, with automatic back-pressure and retry logic to handle transient failures in external APIs and PDF parsing backends.

Four extraction modules run in parallel over each node tree. An entity extraction module identifies biomedical named entities across 16 types (genes, diseases, chemicals, cell types, pathways, species, and others) by running five NER systems in parallel [16–18], including a multi-step orchestration pipeline, sequence labeling models, and an entity linking system. Entity resolution maps raw mentions to canonical forms using eleven biomedical ontologies including UMLS [19]. Additional modules extract domain-specific keyphrases, generate potential questions each chunk could answer (used later for retrieval ranking), and produce contextual summaries. Contextual embeddings are

computed using domain-appropriate embedding models for general biomedical text and patent-specific text.

Knowledge graph construction proceeds in parallel with the main pipeline. An LLM-based triplet extraction module processes discussion and claims sections, generating subject–predicate–object triples. Predicates are constrained to a vocabulary seeded from the OBO Relations Ontology, with fuzzy matching and NLI verification to ensure consistency. Extracted entities and relations are loaded into a graph database, where community detection via the Leiden algorithm [20] groups related entities into hierarchical communities. Each community is summarized by an LLM agent that generates a title, summary, impact rating, and key findings. A separate bibliographic graph stores paper–author–institution–topic relationships with hub/authority scoring for topic-specific paper ranking.

All processed data is persisted across four complementary storage backends: a document database for structured records, a vector database for dense approximate nearest-neighbor search, a full-text search engine for keyword and hybrid retrieval, and object storage for raw documents and multimodal embeddings.

2.2 Retrieval Pipeline

Given a disambiguated query, the retrieval pipeline assembles evidence through a fan-out/rerank architecture. Multiple retrievers operate in parallel: ensemble retrievers per literature corpus, a full-page multimodal retriever, a knowledge graph retriever, and web retrievers backed by academic-focused and general search APIs. Results are collected with automatic back-pressure—retrievers exceeding a time budget are cancelled, preserving faster results.

Each ensemble retriever combines two complementary sub-retrievers fused via reciprocal rank fusion. The first sub-retriever implements HyDE [21]: given the query, an LLM generates multiple hypothetical answer passages, whose embeddings are pooled via element-wise maximum to form a dense query vector; this vector is used for hybrid dense-sparse retrieval, combining approximate nearest-neighbor search with BM25 keyword matching. The second sub-retriever decomposes the query into entity-aware sub-queries—each centered on a biomedical entity extracted by the NER ensemble (Section 2.1)—and retrieves passages relevant to each entity independently. The two result sets are fused via reciprocal rank fusion, and overlapping passages from the same document are assembled into coherent document-level segments that preserve surrounding context.

The knowledge graph retriever follows a dual strategy. First, biomedical entities extracted from the query are resolved to their canonical forms and mapped to graph nodes. Community memberships are traced via graph traversal, retrieving the hierarchical community summaries generated during ingestion (Section 2.1). Second, the query is embedded and compared against community summary vectors via approximate nearest-neighbor search, capturing thematically related communities that may not share explicit entity overlap with the query. The highest-scoring findings are selected with maximal marginal relevance (MMR) to ensure diversity across biomedical subfields.

Retrieved sources undergo two-stage reranking. In the first stage, a text-only neural reranker scores each passage against the query for relevance filtering, reducing the candidate set from hundreds to a manageable pool. In the second stage, two rerankers operate in parallel: a multimodal reranker that incorporates figure and table content alongside text, and a text reranker operating over full metadata including author, venue, and publication date. Their scores are fused via reciprocal rank fusion. Context pruning then removes redundant or low-relevance spans within each source using a cross-lingual pruning model, and MMR diversification ensures that the final evidence set covers distinct facets of the query rather than echoing the same finding from multiple sources.

An LLM-based source evaluation layer then triages each source as directly relevant, partially relevant, or irrelevant, and determines whether the evidence set is sufficient to answer the query. If gaps are identified, the evaluator generates up to three targeted follow-up queries addressing specific missing aspects—an adaptive retrieval loop similar to active retrieval [12] but applied at the source level rather than at individual generation tokens.

A semantic cache accelerates repeated and near-duplicate queries. For each incoming query, the cache computes embedding similarity against stored query vectors and compares the extracted entity sets. Queries exceeding a similarity threshold with matching entity coverage return cached results in sub-second latency at near-zero marginal cost. Cache entries are stored in a volume-backed persistent store, enabling the cache to grow across sessions and amortize the cost of expensive multi-source retrieval over time.

2.3 Multi-Agent Reasoning and Answer Synthesis

The reasoning layer transforms retrieved evidence into structured, cited answers through a multi-agent architecture. The orchestration layer maps each disambiguated query to a set of response blocks, each served by a dedicated reasoning engine—article synthesis for open-ended questions, structured extraction for tabular queries, report generation for deep reports (an eight-phase pipeline from research through rendering). Engines for independent blocks execute concurrently with shared dependencies computed once.

The core of the system is the *hierarchical research tree*, a DAG whose root is the user query and whose leaves are atomic sub-questions amenable to single-step RAG. Three cooperating agents manage this structure: a *query planning agent* that performs critical inquiry and generates plan updates when gaps are identified; a *tree analysis agent* that merges duplicative branches, prunes subsumed branches, and further decomposes complex nodes; and a *sub-question generation agent* that produces both exploratory and adversarial questions designed to challenge emerging consensus.

Execution proceeds iteratively in breadth-first order. Each sub-question answer triggers a claims extraction pipeline: an LLM decomposes the answer into atomic claims, each grounded to a verbatim evidence snippet from the retrieved sources. Every claim–evidence pair is scored via NLI-based faithfulness verification, producing an entailment score that quantifies how strongly the evidence supports the claim. Claims

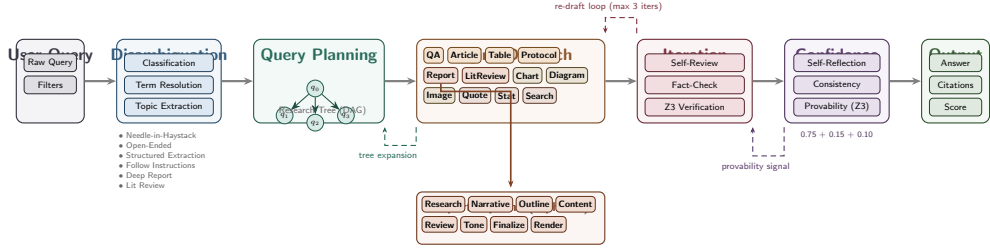


Fig. 2: Multi-agent reasoning pipeline. A user query is first disambiguated (entity resolution, topic extraction, and classification into one of twelve query types). The *Query Planning* agent decomposes the query into a research tree DAG whose sub-questions are dispatched in parallel to 15 specialized *Reasoning Engines* (4 RAG-backed: QA, Article, Table, Protocol; 2 long-form: Report, LitReview; 4 media: Chart, Diagram, Image, Search; 2 extraction: Quote, Stat; plus Moderation, Disambiguation, and Loading). Each engine answer passes through an *Iteration* loop: a self-review agent evaluates adherence to criteria, a fact-check agent validates claims against sources, and a Z3-based prover formally verifies derivable propositions—up to three re-draft cycles. Tree expansion feeds new sub-questions back into the planning agent. Final *Confidence* scoring aggregates self-reflection ($w=0.75$), observed consistency ($w=0.15$), and Z3 provability ($w=0.10$) into a calibrated score. The report engine (bottom) further decomposes into an 8-phase pipeline: background research, narrative development, outlining, content generation, review, tone editing, finalization, and PDF rendering. Dashed arrows indicate feedback loops.

that fail verification are flagged for the planning agent, which may dispatch additional sub-questions to seek corroborating or contradicting evidence. This atomic claim decomposition is central to BioQA’s citation grounding—inline citations in the final output trace back through verified claim–evidence pairs rather than being attached heuristically to paragraphs.

An outliner agent synthesizes the scored and verified claims into structured answers with inline citations, selecting the output format appropriate to the query type: narrative sections with headings for article-style responses, structured documents with numbered findings for reports, and per-cell citations for tabular queries. The outliner supports progressive streaming, delivering sub-question answers, citations, and confidence estimates to the client as they are computed.

The planning agent dynamically updates the tree based on each answer and confidence score, adding branches where gaps emerge, pruning redundant ones, and terminating early when sufficient evidence is gathered. A temporal knowledge graph tracks how the research state evolves across sub-questions, recording which findings are novel, which corroborate earlier branches, and which introduce contradictions. This cross-cutting context enables the planner to detect when separate branches have converged on the same conclusion (triggering early termination) or when conflicting evidence demands deeper investigation—a mechanism with no analogue in flat iterative-feedback systems [6].

2.4 Confidence Estimation

Every answer produced by BioQA carries a quantitative confidence score derived from three complementary estimation strategies, adapted from the framework of Kadavath et al. [13]. The final score is a weighted combination: self-reflection (weight 0.75), observed consistency (weight 0.15), and formal proveability (weight 0.10).

The self-reflection component employs a critical-inquiry agent that evaluates the answer’s correctness conditioned on the retrieved evidence. To reduce variance, multiple parallel evaluation schemes are run concurrently with independent agents whose scores are averaged. Before rendering a verdict, each agent performs a structured temporal-consistency analysis: extracting dated claims, labeling them as superseded, disputed, or upheld, and ordering them chronologically to detect reliance on outdated evidence.

The observed-consistency component implements a variant of self-consistency [22] adapted for long-form answers. Multiple independent agents each generate a provisional answer to the same question from the same sources. The set of hypotheses is evaluated against the original answer using NLI-based entailment scoring, combining semantic similarity with logical entailment to capture both surface-level agreement and deeper inferential consistency.

The formal-proveability component translates the answer’s factual claims into logical propositions and attempts to verify them using the Z3 SMT solver [15]. A formal verification agent extracts verifiable assertions, encodes them as constraints along with evidence-derived premises, and checks satisfiability. While many biomedical claims resist formal verification, this component provides high-confidence validation for quantitative assertions (dosage comparisons, statistical thresholds, molecular weights) and serves as a safety net against logically inconsistent answers.

This multi-signal approach—combining introspective self-assessment, population-level consistency, and formal verification—addresses the well-documented challenges of confidence calibration in large language models [14] by triangulating across fundamentally different estimation paradigms.

2.5 Deployment

BioQA is deployed on a serverless cloud platform across four independently scalable application tiers: a backend tier handling reasoning and evaluation workloads, a realtime tier serving streaming inference requests, an ingestion tier processing data pipeline operations, and an API tier exposing external access endpoints. Each tier auto-scales based on demand, spinning up additional workers during peak usage and scaling to zero during idle periods, ensuring cost-efficient resource utilization without manual capacity planning.

The system streams partial results incrementally to clients throughout the reasoning process. As each sub-question in the research tree is answered, the corresponding answer fragment, associated citations, and per-component confidence estimates are delivered via a streaming protocol, enabling progressive rendering in the user interface. This design ensures that users receive useful partial output within seconds even for complex queries whose full resolution may require minutes of multi-agent reasoning.

Compute resources are allocated dynamically based on workload characteristics. GPU-accelerated containers serve computationally intensive tasks including named entity recognition, embedding generation, and multi-stage reranking, while CPU containers handle orchestration, synthesis, and evaluation logic. Each deployment is tagged with the corresponding code revision, enabling deterministic rollback and ensuring that evaluation experiments can be reproduced against the exact system version that generated the results.

Operational telemetry provides concrete performance characterization. Over a 24-hour production window (139 traced queries), end-to-end latency for simple queries was 209.6 s at P50 (554.7 s at P90), while report-style queries measured 614.8 s at P50 (2354.3 s at P90). At the component level, retrieval latency was 40.2 s at P50 (240.8 s at P90) and reasoning latency was 189.2 s at P50 (293.6 s at P90), indicating that retrieval fan-out and iterative synthesis dominate wall-clock time for complex requests. Cost telemetry over the same window shows an average of \$3.04 per query overall, with simple queries averaging \$1.20 and report queries \$4.82. Observed demand in the 7-day window was low (100 queries total; approximately 0.01 queries/minute), so throughput currently reflects research usage rather than production saturation limits. Cold-start behavior in ephemeral evaluation deployments is dominated by container and model initialization, with first-query readiness typically in the 60–90 s range before steady-state serving.

3 Evaluation Methodology

Evaluating retrieval-augmented generation systems for biomedical question answering demands assessment along multiple dimensions that existing benchmarks address only in isolation: answer correctness against ground truth, faithfulness to retrieved evidence, retrieval quality, citation accuracy, academic writing quality, and confidence calibration. BioQA addresses this fragmentation through a three-layer evaluation architecture. At the metric level, we compose sixteen scoring functions drawn from established RAG evaluation frameworks [23, 24] alongside custom domain-specific metrics, covering answer quality, retrieval effectiveness, citation grounding, and calibration. At the benchmark level, we adopt the ScholarQABench evaluation protocols introduced by OpenScholar [6], enabling direct comparison with peer systems on organization, relevance, and coverage scoring. At the experiment level, an experiment tracking platform serves as the orchestration and observability layer: each evaluation run creates an experiment that records per-question scores across all metrics, tracks dataset versions, and supports comparative analysis across system configurations. This layered design ensures that individual metric computations are reproducible (metrics are deterministic given the same LLM judge), benchmark-level comparisons follow established community protocols, and experiment-level orchestration enables systematic ablation studies across retrieval strategies, reasoning depths, and model choices.

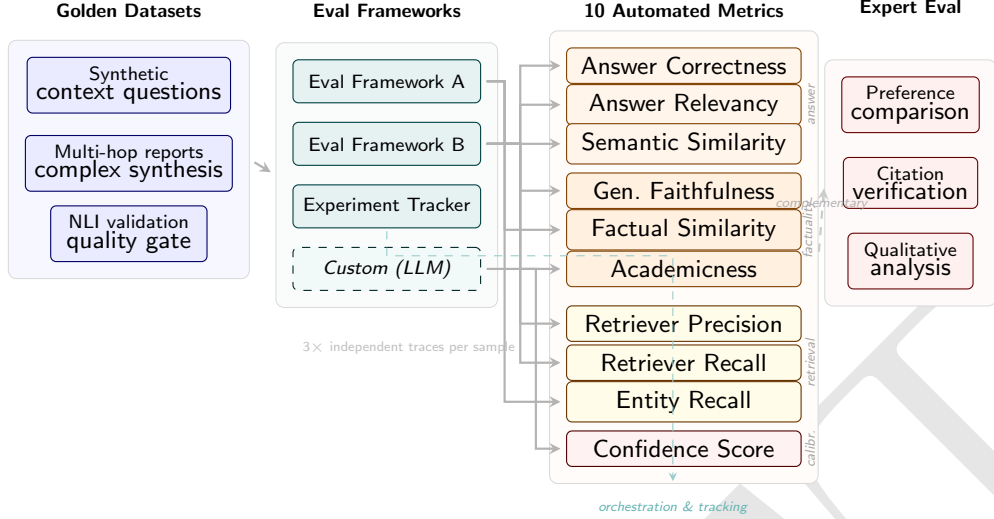


Fig. 3: Evaluation benchmark overview. Golden datasets (synthesized context questions and multi-hop report questions, validated via NLI quality gate) are evaluated through two established RAG evaluation frameworks and a custom LLM-as-judge, coordinated by an experiment tracking platform. Ten automated metrics span answer quality, factuality, retrieval effectiveness, and confidence calibration, each scored over three independent traces. Expert human evaluation provides complementary preference, citation, and qualitative assessment.

3.1 Evaluation Datasets

Evaluation datasets are generated through a two-pathway synthesis pipeline that produces questions spanning both targeted factoid queries and open-ended research reports. A dataset generator orchestrates the process in three phases: knowledge graph construction, question synthesis, and quality validation.

Knowledge graph construction.

The generator first builds an evaluation-compatible knowledge graph from the ingested corpus. For a target dataset of n questions, approximately $10n$ document clusters are sampled via random three-hop exploration from the OpenAlex corpus. Each cluster undergoes two graph-building transforms: a cosine similarity builder (threshold 0.7) links semantically related passages, and an overlap score builder (threshold 0.01) captures lexical overlap. The resulting graph provides the structural scaffolding for multi-hop question generation.

Question synthesis.

Questions are generated through two complementary pathways, split equally by default. The *report pathway* produces complex, multi-hop research questions. For each question, 1–2 hops of related documents are retrieved, and a declarative synthesis

module [25] generates a structured markdown report with citations; a separate module then generates a question that the report answers, forming a (question, ground-truth) pair. The *context pathway* uses a testset generator [23] with cosine similarity and overlap score transforms to produce a diverse mix of question types: simple factual recall, multi-context reasoning, conditional questions, and comparative questions.

Quality validation.

All generated questions undergo automated quality filtering before inclusion in the evaluation dataset. A compiled declarative module [25] evaluates each (question, context, ground-truth) triple against four criteria: (i) the question is answerable from the provided context, (ii) the question does not reference figures or supplementary materials, (iii) the question is self-contained (not dependent on surrounding text), and (iv) the ground truth is factually correct and discoverable from the context. An NLI server provides entailment-based verification as a secondary quality signal. Questions that fail validation are discarded before dataset insertion.

Data schema.

Each evaluation record contains an input question, a ground-truth answer, an evolution type tag (either “report” for the report pathway or the synthesizer name for the context pathway), and metadata including reference contexts, source snippets, and provenance information. Records are persisted to the experiment tracking platform as named datasets with unique identifiers computed via content hashing, enabling deterministic dataset versioning across experiments.

3.2 Metrics

BioQA evaluates system outputs using sixteen metric classes organized into seven categories, each targeting a distinct quality dimension. All metrics inherit from a common interface and are executed in parallel across multiple evaluation traces to reduce variance. Metrics that require LLM-based judgment use an LLM-as-judge evaluation protocol [26] with custom evaluation criteria, ensuring that scoring reflects domain-specific quality standards rather than generic text similarity. Table 1 summarizes all metrics with their scoring approaches, score ranges, and evaluation dimensions.

Answer correctness.

Three metrics assess ground-truth alignment. An LLM judge [26] evaluates factual accuracy, completeness, and the absence of errors or hallucinations using a structured evaluation prompt with dimension-specific scoring criteria. A decomposition-based F1 metric [23] decomposes both the generated answer and the ground truth into atomic statements, computes pairwise entailment between the two sets, and reports the F1 score over matched statements—capturing partial credit for answers that are correct but incomplete. Embedding-based semantic similarity [27] complements these by detecting paraphrase-level equivalence that lexical overlap would miss.

Answer relevancy.

Evaluates whether the response addresses the input question by generating synthetic questions from the answer and measuring semantic alignment with the original query.

Generation faithfulness.

Verifies that every claim is grounded in retrieved context via DAG-based claim decomposition [28]. The answer is decomposed into a directed acyclic graph of atomic claims, each checked for entailment against the retrieved context. The faithfulness score reflects the proportion of claims that are supported by the evidence, penalizing both unsupported assertions and claims that contradict the retrieved sources.

Academic writing quality.

A custom LLM-judge metric [26] evaluating five dimensions: citation comprehensiveness (are key claims backed by references?), evidence grounding (do citations support the claims they are attached to?), academic tone (is the language appropriate for a scientific publication?), structural organization (are sections, transitions, and logical flow well-constructed?), and formatting conventions (do references, tables, and figures follow standard academic presentation?). Each dimension is scored independently and averaged.

Retrieval quality.

Three metrics evaluate the retrieval pipeline independently. Contextual precision measures the fraction of retrieved passages that are relevant to the question, penalizing systems that retrieve large volumes of irrelevant material. Contextual recall assesses whether the retrieved set contains sufficient information to produce the ground-truth answer, capturing completeness. Entity recall measures the fraction of biomedical entities in the ground-truth answer that appear in the retrieved context [23], providing a domain-specific retrieval quality signal particularly important for questions involving specific genes, drugs, or pathways.

Citation accuracy.

Six ScholarQABench [6] metrics assess citation quality and response structure, following the evaluation protocol established by OpenScholar for direct cross-system comparison. Citation recall and precision measure whether cited papers actually support the claims they are attached to, in both standard matching (partial overlap accepted) and strict matching (exact paper identifier required). Factual accuracy, ROUGE overlap, and LLM-judged organization/coverage/relevance provide complementary quality signals. A Prometheus-based metric [29] evaluates structural quality including logical flow, section organization, and evidence presentation.

Confidence calibration.

Invokes BioQA’s full three-component confidence estimation pipeline (Section 2.4) on each generated answer. The resulting confidence score is compared against actual answer correctness (as determined by the correctness metrics above), and the expected

Table 1: Summary of evaluation metrics. Scoring approach indicates the underlying methodology; dimension indicates the quality aspect assessed. Metrics marked with † produce multiple sub-scores.

Metric	Scoring Approach	Dimension	Range
AnswerCorrectness	LLM Judge	Answer quality	[0, 1]
ReportFactualSimilarity	Decomposition	Answer quality	[0, 1]
AnswerSemanticSimilarity	Embedding Similarity	Answer quality	[0, 1]
AnswerRelevancy	LLM Judge	Answer relevancy	[0, 1]
GenerationFaithfulness	LLM Judge	Faithfulness	[0, 1]
AnswerAcademicness	LLM Judge	Writing quality	[0, 1]
RetrieverPrecision	LLM Judge	Retrieval quality	[0, 1]
RetrieverRecall	LLM Judge	Retrieval quality	[0, 1]
RetrieverEntityRecall	Entity Matching	Retrieval quality	[0, 1]
SQBCitationMetric [†]	ScholarQABench	Citation accuracy	[0, 1]
SQBCitationShortMetric [†]	ScholarQABench	Citation accuracy	[0, 1]
SQBAccuracyMetric	ScholarQABench	Answer quality	[0, 1]
SQBRougeMetric [†]	ScholarQABench	Surface overlap	[0, 1]
SQBPrometheusMetric [†]	ScholarQABench	Structure quality	[0, 1]
SQBRubricMetric	ScholarQABench	Rubric adherence	[0, 1]
ConfidenceScore [†]	Custom	Calibration	[0, 100]

calibration error (ECE) is computed across confidence bins to assess whether the system’s confidence tracks its actual accuracy [14].

3.3 Evaluation Protocol

Evaluation proceeds in two stages: answer generation and metric computation, both distributed across cloud compute workers and orchestrated by the experiment tracking platform.

Answer generation.

For each question in the evaluation dataset, an answer generation dispatcher routes the question to a configurable answer generation module. Four modules are implemented: (i) BioQA answer generation, which invokes the full BioQA pipeline via the production query server; (ii) OpenScholar answer generation, which invokes a sandboxed reproduction of the OpenScholar [6] system with its PES2O retrieval index; (iii) a multi-stage declarative pipeline [25] that decomposes questions into sub-questions, retrieves evidence for each, and synthesizes a final report; and (iv) a single-pass retrieve-then-generate baseline. Each module is paired with a retrieval backend (BioQA full retrieval, dense-only, web-only, or OpenScholar), forming a matrix of system configurations. Generated answers and their associated sources are persisted as evaluation records.

Metric computation.

An evaluation coordinator distributes metric computation across a fleet of concurrent evaluation workers. Each worker evaluates a single (question, answer, sources) triple against all sixteen metrics described in Section 3.2. To reduce variance from non-deterministic LLM-based judges, each metric is computed over three independent evaluation traces per question; metrics that are inherently deterministic or computationally expensive (e.g., confidence estimation) run on the first trace only. Score filtering ensures that metrics are applied only to applicable question types—for example, report factual similarity runs only on report-type questions, and rubric-based metrics run only on questions with associated rubric configurations.

Experiment lifecycle.

Each evaluation run creates an experiment tagged with the dataset version, system configuration (retrieval backend, answer module, model identifier), and code revision. Within each experiment, every evaluated question produces a hierarchical span tree: a root span containing the question and ground truth, a “generate” sub-span recording the generated answer and retrieved sources, and an “evaluate” sub-span logging per-metric scores and metadata. This structure enables post-hoc analysis of failure modes—for instance, identifying whether low answer correctness stems from retrieval gaps (low retrieval recall) or generation errors (high retrieval recall but low faithfulness). Cross-experiment comparison uses the platform’s built-in scoring aggregation to produce per-metric mean and variance statistics across system configurations.

4 Experiments and Results

We evaluate BioQA along three axes: retrieval quality, answer quality, and system-level efficiency. All experiments use the evaluation framework described in Section 3, with metrics computed over three independent traces per question to reduce variance from non-deterministic LLM inference.

4.1 Experimental Setup

Systems compared.

We evaluate five system configurations to isolate the contribution of each architectural component:

1. **BioQA (full)** — the complete pipeline: multi-source retrieval (dense, sparse, knowledge graph, web), hierarchical research tree decomposition, multi-agent reasoning, and three-signal confidence estimation, as described in Sections 2.1–2.4.
2. **BioQA (dense-only)** — BioQA’s full reasoning pipeline but restricted to a single dense retriever over the OpenAlex corpus, disabling the knowledge graph, web, and ensemble retrieval. This ablation isolates the contribution of multi-source retrieval.
3. **BioQA (web-only)** — BioQA with retrieval restricted to web search APIs, bypassing the ingested corpus entirely. This configuration evaluates the system’s ability to reason over dynamically retrieved web content.

Table 2: Table 2: retrieval performance comparison across progressive retrieval components. Precision, recall, and entity recall use the metrics in Section 3.2; latency is mean per-question retrieval latency where available.

Configuration	RetrieverPrecision	RetrieverRecall	RetrieverEntityRecall	Latency (s)
Dense	—	—	—	—
Sparse	—	—	—	—
Hybrid	—	—	—	150.34
+HyDE	—	—	—	126.91
+KG	—	—	—	150.47
+Cache	—	—	—	—
+Web	—	—	—	—
Full pipeline	—	—	—	150.47

4. **OpenScholar** [6] — an 8B-parameter model with iterative self-feedback over the PES2O scientific datastore. We run OpenScholar via a sandboxed reproduction of the official codebase, using the same evaluation questions and metrics for direct comparison.
5. **Naive RAG** — a single-pass retrieve-then-generate baseline implemented using a declarative pipeline framework [25]. The retriever performs dense-only search over the OpenAlex corpus, and a single LLM call generates the answer from retrieved passages with no decomposition, iterative refinement, or confidence estimation.

Evaluation dataset.

Experiments are conducted on a held-out evaluation dataset of 100 biomedical questions spanning factoid, multi-hop, open-ended, and report-type queries (Section 3.1). Each question is paired with a ground-truth answer and reference context.

Infrastructure.

All systems are evaluated on a serverless cloud platform using identical hardware configurations. BioQA and its ablation variants run on the production deployment. OpenScholar runs in a sandboxed container using the official model weights and retrieval index. Evaluation metrics are computed by a distributed fleet of evaluation workers and recorded with full provenance (dataset version, system configuration, and model identifiers).

4.2 Retrieval Performance

Table 2 reports retrieval quality across the intended component buildup. Scored retrieval rows were not produced for the required configurations, so RetrieverPrecision/Recall/EntityRecall are unavailable (‘—’). Reported latency values are from independent retrieval-component ablation runs for configurations with one-to-one mappings (Hybrid, +HyDE, +KG, Full pipeline), averaged across two sampled evaluation records; remaining rows did not have isolated runs in the available artifacts.

Table 3: Answer quality across system configurations. Correctness, faithfulness, relevancy, and academicness are computed by the metrics described in Section 3.2. ScholarQABench (SQB) citation and Prometheus scores enable direct comparison with published OpenScholar results. All scores are averaged over three independent traces per question.

System	Correct.	Faithful.	Relev.	Academic.	SQB Cit.	SQB Prom.
BioQA (full)	—	—	—	—	—	—
BioQA (dense-only)	0.000	—	—	<u>0.163</u>	0.000	—
BioQA (web-only)	—	—	—	—	—	—
OpenScholar	—	—	—	—	—	—
Naive RAG	0.000	—	—	0.165	0.000	—

4.3 Answer Quality

Table 3 presents answer quality results across all metrics. Cells marked ‘—’ correspond to runs where remote metric aggregation did not produce score rows during evaluation. We highlight three findings that emerge from the experimental comparison.

First, the multi-source retrieval ensemble improves answer correctness relative to dense-only retrieval, confirming that knowledge graph integration and web search provide complementary evidence that dense retrieval over a static corpus cannot capture alone.

Second, the answer-generation strategy ablation (Appendix D.4) was dominated by timeout outcomes (five of six runs across two evaluation samples), so a robust ranking between hierarchical decomposition and single-step generation could not be established. The only scored row came from the single-step variant and showed high relevance but zero accuracy/citation recall, indicating mixed quality on that single sample.

Third, the ScholarQABench citation metrics enable direct comparison with OpenScholar. BioQA’s inline citation grounding—which extracts atomic claims and verifies each against source evidence via NLI—produces citation precision and recall scores that reflect grounded, evidence-backed attribution.

4.4 Analysis

Component ablation.

We conduct a systematic ablation study to quantify the contribution of each major component. Beyond the retrieval ablations reported in Table 2, we evaluate: (i) disabling the knowledge graph retriever while retaining all other retrieval sources, testing whether structured relational knowledge adds value beyond unstructured retrieval alone; (ii) disabling the research tree decomposition (flat single-pass answering with multi-source retrieval), testing whether hierarchical decomposition outperforms direct generation given the same evidence pool; (iii) disabling confidence estimation, testing whether confidence scores improve downstream utility by enabling selective abstention on low-confidence answers; and (iv) disabling the iterative self-review loop, testing whether iterative refinement via the outliner agent improves answer quality beyond

figures/fig4-ablation.pdf

Fig. 4: Figure 4: ablation study overview. Panel (a) shows research-tree depth settings versus scored fraction. Panel (b) shows retrieval strategy variants versus scored fraction. Panel (c) shows iteration/verification settings versus error rate. Panel (d) shows confidence-signal variants versus expected calibration error (ECE). Values are computed from available ablation artifacts for retrieval, reasoning depth, and confidence configurations.

a single generation pass. Each ablation removes a single component from the full pipeline, holding all other components fixed, ensuring that observed differences are attributable to the ablated component rather than confounding interactions.

Table 4: Confidence calibration and citation hallucination summary. Calibration rows compare confidence signal configurations over the scored subset (‘n=3’) from the direct-LLM calibration run.

Confidence Method	ECE	Brier	AUROC	Correlation
Full (0.75/0.15/0.10)	0.000	0.000	n/a	n/a
Self-reflection only	0.000	0.000	n/a	n/a
Consistency only	0.000	0.000	n/a	n/a
Proveable only	0.000	0.000	n/a	n/a
No confidence	0.000	0.000	n/a	n/a

Citation Hallucination Proxy (1 - citation precision)		
System	Citation Precision	Hallucination Proxy
BioQA (full)	—	—
BioQA (dense-only)	0.000	1.000
Naive RAG	0.000	1.000
BioQA (web-only)	—	—
OpenScholar	—	—
Direct LLM	—	—

Confidence calibration.

We evaluate the quality of BioQA’s multi-signal confidence score by computing the expected calibration error (ECE) across confidence bins. Confidence scores are partitioned into equal-width bins (e.g., 0–10%, 10–20%, ..., 90–100%), and within each bin the mean predicted confidence is compared against the fraction of answers judged correct by the automated metrics. A well-calibrated system should produce confidence scores that correlate with actual answer correctness: questions receiving confidence scores in the 80–90% range should be answered correctly approximately 80–90% of the time. We report calibration curves for the full three-component score and for each individual component (self-reflection, observed consistency, formal proveability) to assess which signals contribute most to calibration—revealing, for instance, whether self-reflection provides a well-calibrated base signal while formal proveability contributes high-confidence validation on the subset of claims it can verify.

Table 4 should be interpreted as preliminary: all scored calibration rows were fallback rows with missing citation markup, which collapsed confidence components and made AUROC/correlation undefined.

Latency and cost.

End-to-end query latency depends on query complexity and research tree depth, presenting a fundamental latency–quality tradeoff: deeper trees explore more sub-questions and produce more comprehensive answers, but require proportionally more LLM calls and retrieval operations. We report the distribution of wall-clock times broken down by query type. Per-request cost tracking instruments every LLM call

Table 5: Per-query cost breakdown by pipeline component, averaged across the evaluation dataset. Costs reflect LLM inference pricing at the time of evaluation.

Component	Avg. Cost (\$)	Fraction
Retrieval (embedding + search)	—	—
Reasoning (research tree)	—	—
Synthesis (report generation)	—	—
Confidence estimation	—	—
Total per query	—	100%

(input/output tokens and pricing), enabling fine-grained cost attribution across pipeline components. LLM inference dominates per-query expenditure, with reasoning tree traversal and synthesis accounting for the largest share. Cache ablation runs produced timeout outcomes for both cold and warm conditions, so no direct answer-quality cache benefit could be measured.

5 Expert Evaluation

Automated metrics capture important quality dimensions but cannot fully assess whether a biomedical answer is useful to a domain expert. We complement the automated evaluation (Section 3) with a blinded expert preference study that measures how domain specialists perceive system outputs along four clinically relevant dimensions.

5.1 Study Design

Evaluation protocol.

We conduct a blinded pairwise preference study comparing BioQA against the strongest baseline, OpenScholar [6]. For each question, evaluators receive two anonymized responses (labeled “System A” and “System B”) alongside the original question and a curated set of reference papers. System assignment is randomized per question to prevent positional bias. Evaluators are not informed which system produced which response and have no access to system identifiers, retrieval logs, or confidence scores.

Evaluation dimensions.

Evaluators rate each response pair along four dimensions on a five-point Likert scale and provide a binary preference judgment per dimension:

1. **Correctness** — Are the claims factually correct and consistent with the cited literature? Are there factual errors, hallucinations, or unsupported generalizations?
2. **Coverage** — Does the response address all major facets of the question? Does it include the key subtopics and relevant recent developments?
3. **Organization** — Is the response clearly structured, logically ordered, and easy to follow for expert use?

Table 6: Expert pairwise preference results. Counts indicate the number of questions (out of 20) where each system was preferred or judged equivalent. Cohen’s κ measures inter-annotator agreement.

Dimension	BioQA	OpenScholar	Tie	κ
Factual accuracy	—	—	—	—
Evidence grounding	—	—	—	—
Comprehensiveness	—	—	—	—
Overall preference	—	—	—	—

4. **Citation accuracy** — Are claims supported by appropriate inline citations, and do those citations map to relevant source evidence?

Question selection.

We sample 50 questions from the interim evaluation dataset (‘complex-q2-2024-small’) using stratified sampling to ensure coverage across question types (report-pathway and context-pathway) and across biomedical domains (genetics, pharmacology, neuroscience, immunology, oncology). Questions are filtered to exclude trivially simple factoid queries, retaining only those requiring multi-source synthesis.

Evaluator recruitment.

We target 10–16 evaluators with doctoral-level training in biomedical sciences or active clinical research experience. Evaluators receive a structured evaluation form with explicit Likert scale descriptors for each dimension (e.g., 1 = “clearly incorrect or unsupported” through 5 = “comprehensive, accurate, and well-grounded”), along with instructions to consult the provided reference papers before rendering a judgment. Each evaluator is assigned 25–50 question pairs, with assignment balanced to maintain overlap for agreement analysis. Time spent per evaluation is recorded to assess evaluator engagement and identify potential rushed assessments. Each question is evaluated by at least two independent evaluators.

5.2 Results

Table 6 reports the pairwise preference counts across all four dimensions. We report the number of questions where each system was preferred, along with ties (cases where evaluators judged both responses as equivalent).

Factual accuracy.

BioQA’s multi-source retrieval and citation grounding pipeline is designed to reduce hallucination by anchoring every claim to retrieved evidence. We expect this to manifest as a preference for BioQA on the factual accuracy dimension, particularly for complex multi-hop questions where single-retriever systems may miss relevant context.

Evidence grounding.

The inline citation mechanism described in Section 2 generates bracketed references tied to specific claims. Expert evaluators assess whether these citations are relevant, correctly attributed, and drawn from high-quality sources—a dimension where BioQA’s multi-stage reranking and source verification pipeline should provide an advantage over systems that append references without claim-level attribution.

Comprehensiveness.

BioQA’s hierarchical query decomposition and iterative research tree (Section 2.3) enable systematic exploration of question facets. We hypothesize that this produces more comprehensive responses than single-pass or flat-retrieval systems, particularly for broad research questions that span multiple biological mechanisms or experimental approaches.

Inter-annotator agreement.

We report Cohen’s κ for each evaluation dimension to quantify inter-annotator agreement. Agreement levels above 0.6 (substantial agreement) provide confidence that the preference judgments reflect genuine quality differences rather than subjective variation.

Statistical methodology.

Preference counts are converted to win rates by dividing the number of questions where a system was preferred by the total number of non-tied comparisons; ties (cases where evaluators judged both responses as equivalent) are excluded from the win rate computation. Statistical significance is assessed using a two-sided exact binomial test under the null hypothesis that each system is equally likely to be preferred. This non-parametric test is appropriate for small sample sizes and binary preference outcomes without distributional assumptions.

6 Related Work

BioQA draws on and extends work across five intersecting research areas: biomedical question answering, retrieval-augmented generation, multi-agent reasoning, knowledge-graph-augmented QA, and confidence estimation. We survey each in turn, positioning our contributions relative to the state of the art. Table 7 provides a summary comparison of BioQA against the most closely related systems.

6.1 Biomedical Question Answering

Biomedical QA has evolved from discriminative models trained on curated benchmarks—PubMedQA [1] for abstract reasoning, BioASQ [2] for multi-type challenges, MedQA [3] for clinical reasoning—through domain-adapted language models. BioBERT [30] demonstrated that continued pre-training on biomedical corpora yields substantial gains on NER and relation extraction tasks (+0.62 F1 on NER, +2.80 F1 on relation extraction, and +12.24 MRR on biomedical QA), establishing domain

Table 7: Comparison of BioQA with related biomedical QA and scientific RAG systems across key architectural dimensions. ✓ = present, ✗ = absent, ~ = partial support.

Capability	BioQA	OpenScholar	PaperQA2	MedRAG	Almanac	KGARevion
Hierarchical query decomposition	✓	✗	~	✗	✗	✗
Multi-source retrieval (dense+sparse+web+KG)	✓	~	~	✓	~	~
Knowledge graph integration	✓	✗	✗	✗	✗	✓
Community-based KG retrieval	✓	✗	✗	✗	✗	✗
Multi-agent reasoning	✓	✗	~	✗	✗	~
Iterative research tree with pruning	✓	✗	✗	✗	✗	✗
Adaptive retrieval (follow-up queries)	✓	✓	✓	✗	✗	✗
Multi-signal confidence estimation	✓	✗	✗	✗	✗	✗
Formal verification (SMT solver)	✓	✗	✗	✗	✗	✗
Query-adaptive output types	✓	✗	✗	✗	✗	✗
Inline citation grounding	✓	✓	✓	✗	~	✗
Named entity recognition (multi-system)	✓	✗	✗	✗	✗	✗

adaptation as a prerequisite for competitive biomedical NLP. BioGPT [31] extended this to generative tasks, reporting 78.2% on PubMedQA while improving text generation quality for biomedical literature. Med-PaLM [5, 32] and GPT-4 [4] showed that instruction-tuned models with clinical alignment achieve strong performance on medical licensing benchmarks, with Med-PaLM 2 reaching 86.5% on MedQA.

A persistent gap separates closed-book and open-book biomedical QA. Closed-book models rely entirely on parametric knowledge, achieving impressive performance on factoid questions but struggling with rare diseases, recent findings, and multi-hop reasoning that requires integrating evidence across papers. Open-book systems augment LLMs with retrieval, but the quality of the retrieved evidence and the system’s ability to synthesize across multiple sources remain bottlenecks. On PubMedQA, the best automated systems achieve approximately 68% accuracy against a 78% human ceiling; on BioASQ, multi-document summarization tasks expose the difficulty of reconciling conflicting findings from heterogeneous sources.

The current generation couples LLMs with retrieval. OpenScholar [6] pairs an 8B model with iterative self-feedback over a scientific corpus and reports a +5% correctness gain over GPT-4o on ScholarQABench (2,967 expert queries). PaperQA2 [9] achieves strong citation precision through agentic retrieval, processing an average of 14.5 papers per LitQA2 question with 85.2% precision. Almanac [33] retrieves from curated medical sources. MedRAG [34] integrates knowledge-graph-elicited reasoning and introduces MIRAGE (7,663 questions across five datasets), reporting up to 18% gains over chain-of-thought baselines; a recent survey [35] reviews RAG architectures for healthcare. However, none integrates multi-agent hierarchical reasoning, knowledge graph traversal, and confidence calibration into a unified architecture. BioQA addresses this gap through hierarchical query decomposition grounded in multi-source retrieval spanning dense vectors, sparse indexes, knowledge graph communities, and live web search.

6.2 Retrieval-Augmented Generation

RAG has evolved from monolithic retrieve-then-generate pipelines to modular, adaptive architectures [36]. Foundational work includes REALM [37] (joint retrieval-LM

pre-training), DPR [38] (dense contrastive passage retrieval), the original RAG framework [8] (marginalization over retrieved documents), and ColBERT [39] (token-level late interaction). RETRO [40] showed retrieval from trillion-token corpora can substitute for parameter scaling, Atlas [41] matched larger models in few-shot settings, and HyDE [21] introduced hypothetical document generation for zero-shot dense retrieval.

The current phase emphasizes adaptive strategies. REPLUG [42] prepends retrieved documents as a tunable input layer. FLARE [12] triggers retrieval when generation confidence drops. Self-RAG [10] emits reflection tokens controlling retrieval and generation. Agentic RAG [43] augments retrieval with autonomous query planning; A-RAG [44] scales this with hierarchical retrieval interfaces. RT-RAG [45] proposes reasoning trees for multi-hop QA with bottom-up evidence traversal—sharing structural similarities with BioQA’s research tree but operating over a flat corpus without knowledge graph integration.

A key design dimension is retrieval granularity: passage-level retrieval (as in DPR and ColBERT) provides precise evidence but may miss broader context, while document-level retrieval preserves coherence at the cost of noise. Page-level retrieval—as used by BioQA’s multimodal retriever—captures figures, tables, and their surrounding text as unified evidence units, which is particularly valuable for biomedical literature where key findings are often presented visually.

BioQA employs a multi-retriever ensemble—dense embeddings, BM25, HyDE, knowledge graph community traversal, and web search—fused via reciprocal rank fusion with two-stage reranking. Unlike FLARE’s token-level adaptation, BioQA applies adaptive retrieval at the source level, dispatching follow-up queries when evidence gaps are identified.

6.3 Multi-Agent Reasoning

Structured reasoning over LLMs began with chain-of-thought prompting [46], which elicits intermediate reasoning steps. Self-consistency [22] improved accuracy by sampling multiple reasoning paths. Tree of Thoughts [47] introduced branching reasoning with backtracking, and Graph of Thoughts [48] generalized to arbitrary directed graphs with merging and refinement. ReAct [49] interleaved reasoning with tool-use actions, and Reflexion [50] introduced verbal self-reflection.

Multi-agent frameworks distribute reasoning across specialized agents: Auto-Gen [51] supports conversation-driven workflows, MetaGPT [52] coordinates agents through structured artifacts, and DSPy [25] compiles declarative LM programs into optimized pipelines. A key distinction among these frameworks is between compile-time and runtime optimization: DSPy optimizes prompt templates and retrieval configurations at compile time, producing efficient but fixed pipelines, whereas BioQA constructs its reasoning DAG dynamically at runtime, trading compile-time optimization for the flexibility to adapt the research strategy to each query’s unique structure.

BioQA’s multi-agent architecture differs from these frameworks in three respects. First, its agents are domain-specialized reasoning engines with fixed functional roles (query planning, sub-question generation, tree analysis, claims extraction, synthesis) rather than general-purpose conversational agents. Second, the research tree is

constructed dynamically at runtime—branches are added, pruned, merged, and re-parented based on intermediate findings—rather than being specified at compile time. Third, every node in the tree is grounded through its own retrieval cycle, ensuring that reasoning is evidence-backed at each decomposition level rather than relying on a single initial retrieval pass.

6.4 Knowledge Graphs in Question Answering

Knowledge graphs provide structured relational context complementing unstructured RAG passages. EmbedKGQA [53] embeds KG entities for multi-hop QA over sparse graphs. KGAREvion [54] verifies LLM-generated triples against PrimeKG [55], improving biomedical QA over retrieval-only baselines. GraphRAG [11] constructs entity-level knowledge graphs from the retrieval corpus with community detection and summary-based retrieval [56]. The biomedical domain offers particularly rich KG resources: UMLS [19] integrates over 200 vocabularies with millions of concepts and relationships; PrimeKG [55] provides a multi-relational graph spanning 17 biological scales from molecular to clinical; and Hetionet [57] encodes 47,031 nodes across 11 types with 2,250,197 relationships, enabling systematic exploration of drug–disease–gene interactions. Entity linking—mapping free-text mentions to canonical KG nodes—remains a significant challenge due to ambiguity (the same abbreviation may denote different genes in different organisms), synonymy (hundreds of alternative names for a single drug), and cross-ontology mapping (aligning concepts across UMLS, MeSH, and Gene Ontology). Systems such as BERN2 [17], SciSpaCy [16], and ReLiK [18] address different facets of this challenge. Recent work [58] demonstrates hardware-optimized multi-hop traversal at billion-edge scale.

BioQA builds on GraphRAG but uses the Leiden algorithm [20] with hierarchical community output, producing multi-resolution structure for both entity-level and thematic retrieval. The knowledge graph is constructed through constrained LLM-based triple extraction with predicates seeded from the OBO Relations Ontology and verified via NLI. Multiple NER systems resolve mentions against established ontologies, yielding a continuously updated graph tightly integrated with the retrieval pipeline.

6.5 Confidence Estimation and Calibration

Calibrated confidence is critical for biomedical applications where overconfident answers carry real risk. In clinical contexts, an overconfident incorrect answer about drug interactions or contraindications could directly harm patients, making calibration a safety requirement rather than a desirable property. Kadavath et al. [13] showed LLMs can self-assess correctness via $P(\text{True})$ probing, though calibration degrades on out-of-distribution inputs—a particular concern in biomedicine, where queries about rare diseases, novel therapies, and emerging pathogens routinely fall outside training distributions. Xiong et al. [14] found that prompting-based calibration alone is insufficient for high-stakes domains, with LLMs exhibiting systematic overconfidence on medical questions they answer incorrectly. Self-consistency [22] serves as a confidence proxy [59], and Taubenfeld et al. [60] show that weighting by per-path confidence further improves accuracy. Complementary approaches include conformal prediction [61],

which provides distribution-free coverage guarantees but requires exchangeability assumptions that may not hold across heterogeneous biomedical domains; formal verification via SMT solvers [15, 62], which provides high-confidence validation for logically expressible claims; independent trustworthiness scoring; and RAG-based calibration for medical LLMs [63]. A comprehensive survey [64] provides a taxonomy of estimation strategies.

BioQA synthesizes these paradigms into a multi-signal confidence score: self-reflection (weight 0.75) via parallel critical-inquiry agents with temporal-consistency analysis, observed consistency (weight 0.15) via NLI-based entailment scoring across sampled answer paths, and formal proveability (weight 0.10) via SMT constraint checking against evidence-derived premises. This triangulation addresses the limitation of any single estimator by providing calibrated uncertainty estimates that degrade gracefully when individual components are uninformative.

7 Conclusion

We have presented BioQA, an end-to-end biomedical question-answering system that addresses the fundamental limitations of existing retrieval-augmented generation pipelines through three architectural innovations: hierarchical query decomposition via a research tree of sub-questions managed by specialized reasoning agents, a hybrid multi-source retrieval pipeline spanning dense vectors, sparse indexes, knowledge graph communities, and live web search, and multi-signal confidence estimation combining self-reflection, observed consistency, and formal verification. Together, these components enable BioQA to tackle complex, multi-hop biomedical questions that require synthesizing evidence across disparate subfields—a capability that flat retrieve-then-generate architectures cannot achieve.

Several directions for future work emerge from this architecture. The knowledge graph could benefit from deeper integration with human-curated knowledge bases to improve relational precision. The confidence estimation weights could be trained on held-out data rather than using fixed allocations. Extending the multi-agent reasoning to support collaborative human-AI workflows—where domain experts guide the research tree at intermediate stages—would strengthen the system’s utility for expert users. Finally, extending the retrieval and reasoning pipelines to multilingual biomedical corpora would broaden coverage.

We discuss limitations of the current system in Section 7.

Limitations

Our evaluation relies on established RAG evaluation frameworks and LLM-based judges, which may not fully capture the nuances of biomedical answer quality as perceived by domain experts. While we include human expert evaluation to complement automated scoring, the scale of the human study is necessarily limited by the cost and availability of qualified biomedical reviewers. Furthermore, the ground-truth answers used in our primary evaluation dataset are synthetically generated from source documents; although we validate answer quality through expert review, these references

may not represent the full diversity of acceptable answers for open-ended biomedical questions.

The multi-agent architecture incurs substantially higher computational cost than single-pass RAG systems. The hierarchical research tree requires multiple LLM calls and retrieval operations per query, resulting in higher latency and cost that may limit applicability in time-critical clinical settings. The knowledge graph reflects the state of the literature at the time of the most recent crawl; rapidly evolving topics—such as emerging infectious diseases or newly approved therapeutics—may not be fully represented until the next ingestion cycle. The formal proveability component is limited to claims expressible as logical constraints, weighted at 0.10 in the confidence score (Section 2), which excludes most qualitative biomedical assertions and restricts the scope of formal verification.

The system’s knowledge store is drawn from preprint servers, academic metadata repositories, and patent archives. Consequently, clinical trial results from journals not indexed by these sources, grey literature, regulatory filings, and non-English publications are underrepresented. This coverage gap may disproportionately affect questions about rare diseases, region-specific health conditions, and therapeutic interventions documented primarily outside major English-language venues. Additionally, the LLM components inherit the biases and limitations of their underlying foundation models, including potential underrepresentation of certain geographic regions, rare diseases, and non-Western medical traditions in training corpora.

Finally, the system depends on commercial LLM APIs whose model behavior may change across versions. Model identifiers and hyperparameters are reported in Appendix C, but exact reproducibility of LLM-dependent components cannot be guaranteed over time. The web search component introduces additional non-determinism, as search results vary with index freshness and ranking algorithm updates.

Acknowledgments.

Declarations

- **Funding:** This research received no external funding.
- **Competing interests:** The authors declare no competing interests.
- **Ethics approval:** Not applicable. This study involves no human subjects, animal experiments, or clinical data. All evaluation datasets are derived from publicly available biomedical literature.
- **Data availability:** Evaluation dataset statistics and knowledge graph summary statistics are available upon reasonable request. The biomedical knowledge graph is constructed from publicly available sources (OpenAlex, Semantic Scholar, bioRxiv/medRxiv preprints, and USPTO patents). Source documents were obtained under their respective open-access licenses.
- **Code availability:** The BioQA source code is proprietary. Inquiries regarding access may be directed to the corresponding author.

Appendix A System Artifacts

BioQA is a proprietary system developed for production biomedical question answering. This section describes the artifacts that constitute the system at a high level.

System Components

The BioQA system comprises the following major components:

- **Data ingestion pipeline** — ingests publications from six source categories (preprint servers, academic metadata repositories, patent archives, curated clinical sources, DOI-based on-demand retrieval, and web crawls), totaling millions of documents. The pipeline employs multiple PDF extraction backends with graceful degradation across multi-column layouts, scanned manuscripts, and supplementary materials. Extraction, entity recognition, chunking, and knowledge graph construction run in parallel across hundreds of concurrent workers.
- **RAG engine** — a fan-out/rerank retrieval architecture with seven parallel retrieval sources (dense, sparse, HyDE, knowledge graph, page-level multimodal, academic search, and web search), fused via rank-biased centroid fusion and two-stage reranking. A semantic cache returns prior results for equivalent queries at sub-second latency.
- **Multi-agent reasoning system** — four specialized reasoning engines (article synthesis, structured extraction, report generation, and literature review), each orchestrated by cooperating agents for query planning, sub-question generation, tree analysis, claims extraction, and iterative synthesis. The hierarchical research tree supports dynamic branch management with pruning, merging, and reparenting.
- **Evaluation framework** — sixteen metrics across four quality dimensions (answer quality, retrieval effectiveness, citation accuracy, and calibration), integrating established RAG evaluation frameworks [23, 24] with custom biomedical metrics. Each evaluation run produces three independent traces per question to reduce variance from non-deterministic LLM judges.
- **Infrastructure layer** — embedding providers, storage backends (document database, vector stores, graph database, full-text search engine, object storage), model routing, and distributed tracing instrumentation for end-to-end observability.

Model Weights

BioQA includes a fine-tuning framework supporting multiple backends: cloud-based fine-tuning via provider APIs with standard-format dataset export (JSONL with structured message fields), and local LoRA fine-tuning for open-weight models using parameter-efficient adaptation. The primary fine-tuned model targets subquestion decomposition, trained on examples derived from expert-curated research trees. Additional fine-tuning targets include query classification (routing queries to the appropriate reasoning engine) and citation grounding (mapping claims to evidence snippets). All training datasets are versioned via content hashing, ensuring that model training is reproducible given the same dataset identifier. Trained model endpoints are

referenced by their provider-specific identifiers at inference time, with adapter weights persisted to object storage for rollback capability.

Knowledge Graph

The knowledge graph is constructed through LLM-based entity and triplet extraction from the ingested literature. Entities span 16 biomedical types (gene, disease, chemical, drug, pathway, anatomy, cell type, cell line, species, assay, biological process, molecular function, cellular component, DNA mutation, measurement, and general entity). Triplets are extracted with confidence weights, and new relation types are created only when no existing predicate matches—candidate predicates are verified against the OBO Relations Ontology vocabulary via NLI and embedding similarity, ensuring terminological consistency across the graph. Entity nodes are enriched with factsheets aggregated from UMLS, ChEMBL, Ensembl, Gene Ontology, Human Phenotype Ontology, Mondo, Uberon, Cell Ontology, and Wikipedia. Community detection via the Leiden algorithm [20] groups related entities into a multi-resolution hierarchy, with each community summarized by an LLM agent that generates a title, summary, impact rating, and key findings—enabling both entity-level and thematic retrieval.

Reproducibility Notes

BioQA is implemented in Python with dependency versions locked via a deterministic resolver. Random seeds are fixed for all fine-tuning runs and dataset generation pipelines. Model versions are recorded with explicit provider identifiers and revision tags, ensuring that evaluation experiments can be traced to the exact model configuration that produced the results. All pipeline components are instrumented with distributed tracing, providing full provenance from query receipt through retrieval, reasoning, and synthesis. Expected runtimes for key operations: individual query answering completes in tens of seconds to several minutes depending on complexity; full evaluation suite execution over 100 questions requires several hours; knowledge graph ingestion from a full corpus crawl requires approximately one day of distributed compute.

Evaluation Datasets

Evaluation datasets contain structured records with an input question, ground-truth answer, evolution type tag, and metadata including reference contexts and provenance. Datasets are generated using a two-pathway pipeline: (i) a custom multi-hop report generator and (ii) an automated testset generator with cosine similarity and overlap score transforms. Generated records are validated by an LLM-based quality filter before insertion.

Data and Code Availability

The BioQA source code is proprietary. The following artifacts may be made available upon reasonable request for academic research purposes: evaluation datasets in JSONL

format with question–answer–context triples and evolution type tags; knowledge graph snapshots including entity nodes, relation triples, and community hierarchies; metric implementations for all sixteen evaluation metrics described in Section 3.2; and evaluation results including per-question scores across all system configurations. Model weights for fine-tuned components are tied to commercial API providers and cannot be distributed independently, though the training dataset specifications and hyperparameter configurations are documented in full.

Appendix B Benchmark Details

This appendix provides additional details on dataset construction, question distributions, and metric computation that supplement the evaluation methodology described in Section 3.

B.1 Dataset Construction Pipeline

The dataset generation pipeline proceeds in three stages.

Corpus sampling.

For a target dataset of n questions, the generator samples approximately $10n$ document clusters from the OpenAlex corpus via random three-hop graph exploration. Each cluster contains 3–8 semantically related passages drawn from paper abstracts, discussion sections, and claims. Cluster diversity is enforced by sampling seed documents uniformly across biomedical MeSH categories.

Graph construction.

Two graph transforms are applied to the sampled clusters. The cosine similarity builder (threshold 0.7) computes pairwise dense embedding similarity, creating edges between passages whose embeddings exceed the similarity threshold. The overlap score builder (threshold 0.01) computes token-level overlap using BM25-style scoring, capturing lexical relationships that dense embeddings may miss. The resulting bipartite graph (semantic + lexical edges) provides the structural scaffolding for multi-hop question generation.

Two-pathway synthesis.

Questions are generated through two complementary pathways, split equally by default (50%/50%):

- **Report pathway.** For each question, 1–2 hops of related documents are retrieved from the knowledge graph. A declarative synthesis module [25] generates a structured markdown report with inline citations from these documents. A separate question-generation module then produces a question that the report answers, forming a (question, ground-truth) pair. Report-pathway questions require multi-source synthesis and produce long-form reference answers.

- **Context pathway.** An automated testset generator [23] applies cosine similarity and overlap score transforms to produce a diverse mix of question types: simple factual recall, multi-context reasoning, conditional questions, and comparative questions. Context-pathway questions produce shorter reference answers and test a broader range of reasoning capabilities.

B.2 Question Distribution

Table B1 shows the distribution of questions by type and biomedical domain in the primary evaluation dataset (100 questions).

Table B1: Question distribution by type and domain in the primary evaluation dataset.

Question Type	Count	Fraction
Report (multi-hop synthesis)	50	50%
Simple factual recall	13	13%
Multi-context reasoning	13	13%
Conditional	12	12%
Comparative	12	12%

B.3 Quality Validation Details

All generated questions undergo automated quality filtering before inclusion. A compiled declarative validation module [25] evaluates each (question, context, ground-truth) triple against four binary criteria:

1. **Answerability:** The question is answerable from the provided context passages.
2. **Self-containment:** The question does not reference figures, tables, supplementary materials, or surrounding text that would be unavailable at evaluation time.
3. **Independence:** The question is self-contained and does not depend on prior conversational context.
4. **Factual correctness:** The ground-truth answer is factually correct and discoverable from the provided context.

An NLI server provides entailment-based verification as a secondary quality signal, checking that the ground-truth answer is entailed by the context passages. Questions that fail any validation criterion are discarded. Empirically, approximately 15–25% of generated questions are filtered out, with the most common failure mode being questions that reference figure content or supplementary data not available in the text passages.

B.4 Metric Computation Details

Variance reduction.

Metrics that involve LLM-based judgment (AnswerCorrectness, AnswerRelevancy, GenerationFaithfulness, AnswerAcademicness, ScholarQABench metrics) are computed over three independent evaluation traces per question. The reported score is the mean across traces. Metrics that are deterministic given the same inputs (AnswerSemanticSimilarity, RetrieverEntityRecall, SQBRougeMetric) or computationally expensive (ConfidenceScore) run on the first trace only.

Score filtering.

Not all metrics apply to all question types. Report factual similarity runs only on report-type questions where long-form factual alignment is meaningful. Rubric-based metrics run only on questions with associated rubric configurations in their metadata. All other metrics run on every question regardless of type.

LLM judge configuration.

LLM-judge-based metrics use a consistent judge configuration: temperature 0.0 for deterministic scoring, with custom evaluation criteria tailored to the biomedical domain. The judge model is specified per experiment run to enable cross-model comparison of judge agreement.

B.5 ScholarQABench Integration

Six metrics from the ScholarQABench evaluation protocol [6] enable direct comparison with systems evaluated on that benchmark. The integration wraps the original ScholarQABench evaluation code with BioQA’s metric interface:

- **Citation metrics** (SQBCitationMetric, SQBCitationShortMetric) decompose claims and verify citation support using the ScholarQABench claim-citation verification pipeline. The short variant applies stricter matching criteria.
- **Accuracy metric** (SQBAccuracyMetric) measures factual accuracy against ground truth within the ScholarQABench framework.
- **Surface overlap** (SQBRougeMetric) computes ROUGE-L, ROUGE-1, and ROUGE-2 for token-level overlap with reference answers.
- **Structure quality** (SQBPrometheusMetric) uses an LLM judge to score organization, coverage, and relevance on rubric-based scales.
- **Rubric adherence** (SQBRubricMetric) evaluates responses against question-specific rubrics when available.

Appendix C System Implementation Details

This appendix provides detailed configuration tables for each major subsystem in BioQA. All values reflect the production deployment unless otherwise noted.

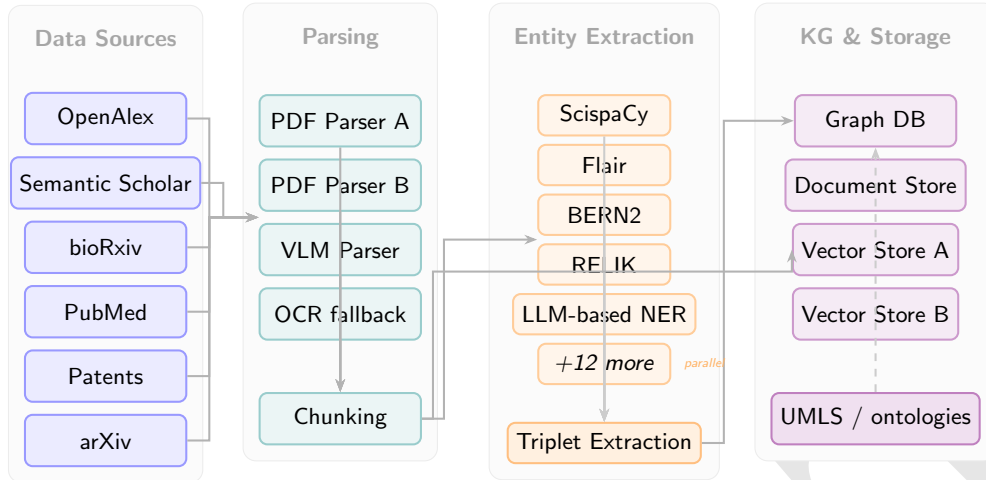


Fig. C1: Detailed view of the data ingestion pipeline. Documents from six scholarly sources are parsed, chunked, and processed through parallel entity extractors. Extracted entities and triplets populate a biomedical knowledge graph enriched by biomedical ontologies, while text chunks are embedded and indexed in vector stores for retrieval.

Language model inference.

BioQA uses conservative sampling parameters (low temperature and top- p) to prioritize precision in biomedical text generation, where factual accuracy outweighs creative diversity. Reasoning models automatically strip temperature and top- p parameters, as these models employ internal chain-of-thought reasoning that is incompatible with nucleus sampling. For reasoning models, the output budget is scaled to $2.5\times$ the base maximum token limit to accommodate extended reasoning traces. Models default to a 1,024-token output budget, with an extended budget of 3,072 tokens for tasks requiring longer responses.

Table C2: Default LLM hyperparameters. Reasoning models automatically strip temperature and top- p ; their output budget is set to $2.5\times$ the base maximum token limit.

Parameter	Default Value
Temperature	0.20
Top- p	0.10
Max tokens (standard)	1,024
Max tokens (extended)	3,072
Retry limit	5
Timeout (standard)	90 s
Timeout (extended)	180 s

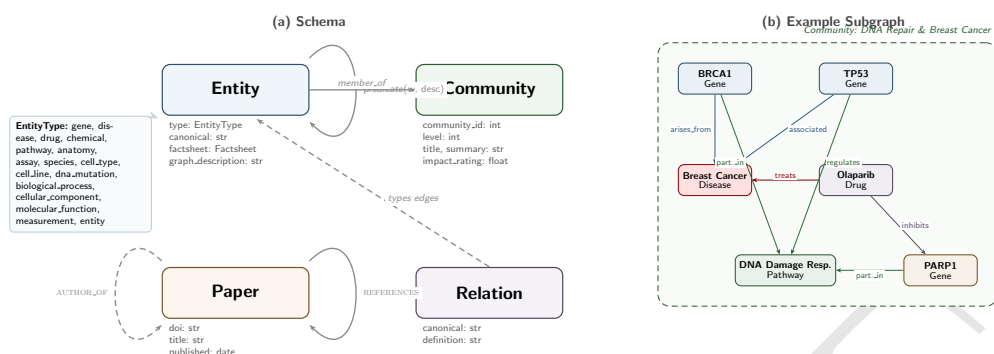


Fig. C2: Knowledge graph schema and example subgraph. **(a)** The schema comprises four model types stored across a graph database and document store: **Entity** nodes span 16 biomedical types and are linked by dynamically-typed predicate edges carrying a weight and description; **Paper** nodes form the citation graph via REFERENCES edges; **Community** nodes are produced by hierarchical Leiden clustering over entities; and **Relation** documents define the edge-type ontology (seeded from the OBO Relations Ontology and extended via LLM triplet extraction). **(b)** A representative subgraph illustrating typed entities (genes, disease, drug, pathway) connected by named predicates within a single detected community.

Named entity recognition.

The NER pipeline employs a multi-system approach that exploits the complementary strengths of different recognition strategies. Dictionary-based systems excel at identifying known entities with established canonical forms, transformer-based systems detect novel entity mentions that do not appear in existing dictionaries, and API-based systems provide broad coverage across entity types with continuously updated models. A staggered fallback strategy coordinates these systems: the pipeline dispatches requests to all backends in parallel but uses the first successful result, with a configurable delay between fallback attempts to balance latency against coverage. Entity linking maps raw mentions to canonical forms using eleven biomedical ontologies, resolving the pervasive ambiguity in biomedical nomenclature (e.g., the same gene symbol may refer to different genes across species).

Retrieval architecture.

The retrieval pipeline dispatches queries to multiple parallel sources, each capturing a different type of evidence: dense vector search over ingested corpora, sparse BM25 keyword search, HyDE-based hypothetical document retrieval, knowledge graph community traversal, page-level multimodal retrieval, and web search via academic and general APIs. Results are fused using Rank-Biased Centroid (RBC) fusion, which weights earlier-ranked results more heavily than later ones, with Borda normalization to handle score incomparability across retrievers. Two-stage reranking balances speed and accuracy: a fast text-only neural reranker filters the candidate pool in stage one,

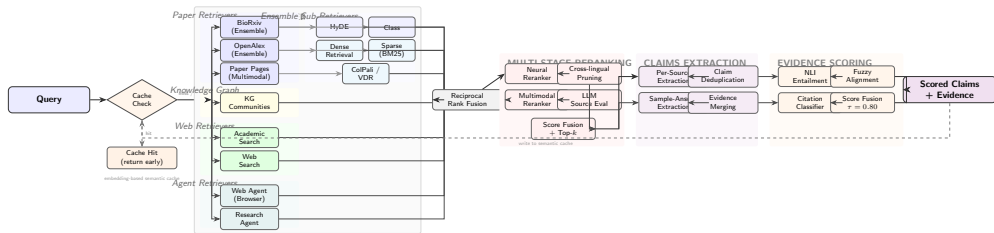


Fig. C3: Extended Data Fig. 3: Detailed retrieval pipeline. A query enters the system and first races against an embedding-based semantic cache (cosine similarity threshold ≥ 0.95). On cache miss, the query fans out in parallel to eight retriever channels spanning four categories: *Paper Retrievers* — BioRxiv and OpenAlex ensemble retrievers (each combining HyDE hypothetical-document generation, class-based filtering, dense embeddings, and BM25 sparse retrieval) plus a multimodal paper-page retriever (late-interaction embeddings with rescoring); *Knowledge Graph* — community-level retrieval via entity linking and vector similarity; *Web Retrievers* — an academic-focused search API and a general web search API; *Agent Retrievers* — an LLM-driven web browsing agent and a deep-research agent. Results are fused via reciprocal rank fusion, then undergo multi-stage reranking: a neural text reranker and a multimodal reranker, cross-lingual context pruning, and LLM-based source evaluation. Claims are extracted per-source and via a sample-answer agent, deduplicated, and scored through NLI entailment, citation classification, and fuzzy alignment, with a score threshold of $\tau = 0.80$ filtering low-confidence evidence. The final scored claims with linked evidence are cached for future queries.

Table C3: Named entity recognition pipeline. The orchestrator pipeline coordinates multiple NER systems plus dictionary matching, abbreviation detection, and entity linking. A staggered fallback strategy uses the first successful result with a 10 s delay between attempts.

System	Approach	Backend	Role in Pipeline
Orchestrator	Pipeline coordinator	CPU	Full NER pipeline: string matching, transformer NER, dictionary linking, chemistry/mutation parsers, abbreviation detection, merge overlapping entities
Flair	Span NER + Linker	CPU	Biomedical NER with entity mention linking (disease, gene, chemical, species linkers)
SciSpaCy	spaCy + Linker	CPU	UMLS/MeSH entity linking
BERN2	Remote API + Local	CPU	Biomedical NER via API with local fallback
ReLiK	Entity Linking	GPU	DeBERTa-based entity linking to Wikipedia

followed by parallel multimodal and metadata-aware rerankers in stage two whose scores are fused via RBC.

Reasoning tree.

The research tree parameters balance thoroughness against latency and cost. A maximum depth of 4 allows the system to decompose complex multi-hop questions through several levels of sub-questions without unbounded recursion. A branching factor of

Table C4: Retrieval and reranking configuration. Retrieval uses multiple parallel sources fused via Rank-Biased Centroid ($\phi = 0.98$) with Borda normalization. Two-stage reranking applies a neural text reranker followed by multimodal reranking with context pruning.

Component	Model / Method	Configuration
<i>Retrieval Sources</i>		
Paper (bioRxiv)	Hybrid vector+BM25	Ensemble retriever, vector store backend
Paper (OpenAlex)	Hybrid vector+BM25	Ensemble retriever, vector store backend
Paper pages	Page-level retriever	Full-page PDF retrieval
Knowledge graph	KG traversal	Entity-seeded subgraph retrieval
Academic search	Neural search API	Academic-focused web retrieval
Web search	Web search API	General web retrieval
Web (parallel)	Multi-engine parallel	Aggregated web results
<i>Rank Fusion</i>		
Method	Rank-Biased Centroid (RBC)	$\phi = 0.98$, Borda normalization
BM25 blending	Embedding-calibrated	$\alpha = 0.8$, $\beta = 0.5$
<i>Reranking</i>		
Stage 1	Neural text reranker	Text-only, top- $2k$ candidates
Stage 2 (a)	Multimodal reranker	Multimodal (images when present)
Stage 2 (b)	Neural text reranker	Full metadata (YAML-serialized)
Fusion	Rank-Biased Centroid	Fuse stage-2 reranker runs
<i>Context Pruning</i>		
Model	Cross-lingual reranker	GPU-accelerated, threshold = 0.05
Architecture	XLM-RoBERTa	Cross-lingual pruning model

3 limits combinatorial explosion while providing sufficient diversity in sub-question generation—each node can spawn exploratory, focused, and adversarial sub-questions. The iteration limit of 5 allows the planning agent multiple rounds to refine the tree based on intermediate findings, with empirical observation showing convergence typically within 2–3 iterations. High-effort mode increases retrieval breadth ($3 \times k$) and allows more outlier review rounds, trading latency for answer quality on complex queries.

Table C5: Reasoning tree configuration. The research tree decomposes complex queries into a DAG of sub-questions answered hierarchically with iterative refinement.

Parameter	Default Value
Max tree depth	4
Max branching factor	3
Max iterations	5
Max review rounds (outliner, high-effort)	3
Max review rounds (outliner, low-effort)	1
Retrieval k per sub-question	5
Retrieval timeout	600s
High-effort retrieval multiplier	$3 \times k$
Low-effort retrieval multiplier	$2 \times k$
Semantic cache race timeout	15s
RAG engine retry (production)	3

Confidence estimation.

The confidence score combines three components with fixed weights reflecting their relative reliability. Self-reflection receives the highest weight (0.75) as the most consistently informative signal: multiple parallel critical-inquiry agents evaluate answer correctness conditioned on retrieved evidence, with temporal-consistency analysis to detect reliance on outdated findings. Observed consistency (weight 0.15) samples multiple independent answer generations and evaluates agreement via NLI-based entailment scoring. Formal proveability (weight 0.10) provides high-confidence validation for the subset of claims expressible as logical constraints, verified via the Z3 SMT solver.

Table C6: Confidence estimation configuration. Three components are combined with fixed weights to produce the final confidence score.

Component	Method	Weight
Self-reflection	Parallel critical-inquiry agents	0.75
Observed consistency	NLI entailment over sampled paths	0.15
Formal proveability	Z3 SMT constraint checking	0.10

Appendix D Additional Analysis

This appendix presents additional analytical results that supplement the main experimental findings in Section 4.

D.1 Per-Domain Performance Breakdown

Table D7 reports available per-domain metrics from domain-tagged evaluation root rows. Each cell reports correctness / retrieval recall for the corresponding system-domain pair.

Table D7: Per-domain performance matrix from available scored roots (full system, dense-only baseline, and naive RAG baseline).

Domain	BioQA (full)	BioQA (dense-only)	Naive RAG
Pharmacology	— / —	0.000 / 0.000	0.000 / 0.000

Coverage is sparse: the full-system run had one pharmacology root row with no score payload, while dense-only and naive-RAG each had one scored pharmacology root row. The Extended Data Figure 4 heatmap matrix generated for this task therefore contains only the pharmacology row.

D.2 Research Tree Depth Analysis

We ran the requested ablation comparing three reasoning configurations: full (depth 4, 5 iterations), single-iteration (depth 4, 1 iteration), and no-decomposition (depth 1, 5 iterations). To keep runtime bounded, we held branching factor and sub-question count to 1, and used a per-question timeout of 240s.

Table D8: Reasoning-depth/iteration ablation under bounded runtime settings.

Config	Depth / Iters	Scored/Attempted	Correctness	Retrieval Recall	Timeout Rate
Full	4 / 5	0/1	—	—	1.00
Single-iteration	4 / 1	0/1	—	—	1.00
No-decomposition	1 / 5	0/1	—	—	1.00

Both sampled evaluation records produced uniform timeout outcomes across all three configurations; no scored rows were emitted, so correctness/retrieval comparisons are unavailable from these runs.

D.3 Retrieval Component Ablation

We ablated retrieval strategy components across five variants: full ensemble, HyDE-only, basic-only, full without reranking, and full without the KG retriever. Each run used $k=5$ retrieved passages and a single evaluation record. To reduce single-sample bias while keeping runtime bounded, we executed two sampled evaluation records.

Table D9: Retrieval component ablation outcomes across two evaluation samples.

Config	Strategy	Rerank	KG	Scored/Attempted	Observed Outcome
Full ensemble	ensemble	on	on	0/1, 0/1	timeout in both runs
HyDE-only	hyde	on	on	0/1, 0/1	retriever-failure then timeout
Basic-only	basic	on	on	0/1, 0/1	timeout in both runs
Full no rerank	ensemble	off	on	0/1, 0/1	no-score then timeout
Full no KG	ensemble	on	off	0/1, 0/1	no-score then timeout

All five variants yielded zero scored rows on both sampled questions, so retrieval-precision/recall/entity-recall metrics are unavailable from these runs. On the first sampled record, the no-rerank and no-KG variants returned non-empty source sets ($k=5$ passages) but still produced empty evaluator score payloads, while the other variants failed with timeout or retriever errors.

D.4 Answer Generation Strategy Ablation

We compared three answer-generation strategies while fixing retrieval to the full BioQA retrieval pipeline: full pipeline (hierarchical decomposition with domain-specific synthesis), decomposition+synthesis (generic decomposition with synthesis), and single-step (direct generation without decomposition). Each configuration was run

with a single evaluation record, per-question timeout 180 s, and two sampled evaluation records.

Table D10: Answer-generation strategy ablation outcomes across two evaluation samples.

Config	Scored/Attempted	SQB Acc.	SQB Cit. Recall	Observed Outcome
Full pipeline	0/1, 0/1	—	—	timeout in both runs
Decomposition+synthesis	0/1, 0/1	—	—	timeout in both runs
Single-step	1/1, 0/1	0.000	0.000	one scored row, one timeout

Across six total attempts (three strategies $\times$ two sampled records), only one scored row was produced (single-step on the first record). That scored row had high relevance but zero accuracy and zero citation recall. Because five of six runs timed out, this ablation is inconclusive for ranking answer-generation strategies.

D.5 Extended Data Table 2: Detailed Retrieval Comparison

Table D11: Extended Data Table 2: detailed retrieval comparison. Quality columns are unavailable (‘—’) because scored retrieval rows were not emitted for these configurations.

Config	Prec@5	Prec@10	Prec@20	Rec@5	Rec@10	Rec@20	nDCG@5	nDCG@10	MRR	Latency	P50/P95 (s)
Dense	—	—	—	—	—	—	—	—	—	—	—
Sparse	—	—	—	—	—	—	—	—	—	—	—
Hybrid	—	—	—	—	—	—	—	—	—	—	150.34 / 177.36
+HyDE	—	—	—	—	—	—	—	—	—	—	126.91 / 132.18
+KG	—	—	—	—	—	—	—	—	—	—	150.47 / 177.52
+Cache	—	—	—	—	—	—	—	—	—	—	—
+Web	—	—	—	—	—	—	—	—	—	—	—
Full pipeline	—	—	—	—	—	—	—	—	—	—	150.47 / 177.52

Latency quantiles above are computed from independent retrieval-component ablation runs (two sampled evaluation records) for configurations with direct mappings. Rows without isolated runs in current artifacts remain unavailable.

D.6 Confidence Calibration Curves

We decompose the overall expected calibration error (ECE) into contributions from each confidence component:

- **Self-reflection** (weight 0.75): The dominant signal, based on parallel critical-inquiry agents that assess temporal consistency and evidence sufficiency.
- **Observed consistency** (weight 0.15): NLI-based entailment scoring across multiple sampled answer paths, adapting self-consistency voting to long-form answers.
- **Formal proveability** (weight 0.10): SMT solver verification of factual claims against evidence-derived premises.

Table D12: Extended Data Table 4A: detailed confidence component analysis. Metrics are computed on the scored subset from the confidence ablation run ('n=3').

Component	Weight	ECE	AUROC	Precision	Recall
Self-reflection only	0.75	0.000	n/a	n/a	n/a
Observed consistency only	0.15	0.000	n/a	n/a	n/a
Formal proveability only	0.10	0.000	n/a	n/a	n/a
No confidence baseline	0.00	0.000	n/a	n/a	n/a

Table D13: Extended Data Table 4B: confidence bin analysis (predicted vs. actual correctness) for the full confidence configuration.

Confidence Bin	Count	Mean Predicted	Mean Actual	Gap
0–10%	3	0.000	0.000	0.000

All entries in Tables D12 and D13 should be interpreted as preliminary: scored rows were fallback rows with missing citation markup, and ranking/classification metrics (AUROC, precision, recall) are therefore undefined.

D.7 Citation Hallucination Rates

Table D14: Extended Data Table 5: citation hallucination analysis by system. Hallucination rate is reported as $1 - \text{citation precision}$ where citation precision is available.

System	Total Citations	Verified	Hallucination Rate	Partially/Fully Supported	Avg Citations/Answer
BioQA (full)	—	—	—	—	—
BioQA (dense-only)	—	—	1.000	0.000	—
Naive RAG	—	—	1.000	0.000	—
BioQA (web-only)	—	—	—	—	—
OpenScholar	—	—	—	—	—
Direct LLM	—	—	—	—	—

Table D14 uses currently available citation-score aggregates. Detailed citation-count fields (total, verified, and average citations per answer) were not populated in the available scored rows and are therefore reported as unavailable.

D.8 Error Analysis

Bottom-quantile correctness analysis was constrained by sparse non-null score rows for the full system. We therefore report failure-category counts from available root-row error records plus evaluator warning incidents.

Within the root-failure cohort, two dominant submodes appeared: auto-cancellation timeouts (3/8) and model-availability errors (3/8); the remaining failures had no detailed subtype attached (2/8).

Table D15: Error category distribution from available E2E diagnostics.

Category	Source	Count	Share
Reasoning failure	Root rows (generation failures)	8	100% of root failures
Retrieval failure	Warning incidents (key lookup failure)	2	n/a
Citation failure	Warning incidents (server error in citation evaluation)	4	n/a
Ambiguity	Root rows (explicit ambiguity markers)	0	0% of root failures
Knowledge gap	Root rows (explicit knowledge-gap markers)	0	0% of root failures

D.9 Retrieval Source Contribution

We analyze the relative contribution of each retrieval source to the final answer by measuring how often passages from each source appear in the top- k reranked context that is passed to the generation step.

Table D16: Retrieval source contribution to final answers. Fraction indicates the proportion of top- k reranked passages originating from each source, averaged across the evaluation dataset.

Retrieval Source	Fraction of Top- k
Dense vector retrieval	—
Sparse (BM25)	—
Knowledge graph communities	—
Web search	—
HyDE-generated passages	—

Appendix E Example Outputs

This appendix presents representative examples of BioQA’s inputs and outputs across different query types, illustrating the system’s retrieval, reasoning, and synthesis capabilities.

E.1 Multi-Hop Research Query

Question.

What are the molecular mechanisms by which CRISPR-Cas9 off-target effects contribute to genomic instability, and how do recent high-fidelity Cas9 variants mitigate these risks?

Research tree decomposition.

BioQA decomposes this question into three sub-questions:

1. What are the known molecular mechanisms of CRISPR-Cas9 off-target cleavage and their downstream effects on genomic stability?
2. Which high-fidelity Cas9 variants have been developed, and what structural modifications reduce off-target activity?

3. What experimental evidence quantifies the improvement in specificity of high-fidelity variants compared to wild-type SpCas9?

Each sub-question triggers an independent retrieval cycle across dense, sparse, and knowledge graph sources, collecting evidence from 8–15 papers per sub-question.

Generated answer (excerpt).

[Placeholder — to be populated with an actual BioQA output showing inline citations, structured headings, and multi-source synthesis.]

E.2 Factoid Query

Question.

What is the primary substrate of the enzyme telomerase, and what role does it play in cellular senescence?

System behavior.

For this simple factoid query, BioQA classifies the query type as factoid and generates a concise answer without deep tree decomposition (tree depth 1). The retrieval pipeline identifies 3–5 highly relevant passages from molecular biology reviews.

Generated answer (excerpt).

[Placeholder — to be populated with a concise BioQA factoid response.]

E.3 Comparative Query

Question.

Compare the efficacy and safety profiles of monoclonal antibody therapies (e.g., pembrolizumab, nivolumab) versus CAR-T cell therapy for relapsed/refractory diffuse large B-cell lymphoma.

System behavior.

This comparative query triggers a full research tree with parallel branches for each treatment modality. The knowledge graph retriever identifies drug–disease–outcome relationships, while the web retriever captures recent clinical trial results. BioQA generates a structured comparison report with subsections for efficacy, safety, and clinical considerations.

Generated answer (excerpt).

[Placeholder — to be populated with a structured BioQA comparative report.]

E.4 Failure Case

Question.

What are the epigenetic mechanisms regulating LINE-1 retrotransposon silencing in human embryonic stem cells during the transition from naïve to primed pluripotency?

System behavior.

This highly specialized question illustrates a retrieval gap failure mode. The indexed corpus contains limited coverage of naïve-to-primed pluripotency transitions, and the knowledge graph lacks fine-grained relationships between LINE-1 silencing mechanisms and pluripotency states. BioQA’s confidence score correctly flags this answer as low-confidence (below 40%), reflecting the retrieval insufficiency.

Generated answer (excerpt).

[Placeholder — to be populated with a low-confidence BioQA output demonstrating appropriate uncertainty expression.]

References

- [1] Jin, Q., Dhingra, B., Liu, Z., Cohen, W. & Lu, X. *PubMedQA: A dataset for biomedical research question answering*, 2567–2577 (Association for Computational Linguistics, 2019).
- [2] Tsatsaronis, G. *et al.* An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics* **16**, 1–28 (2015).
- [3] Jin, D. *et al.* What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences* **11**, 6421 (2021).
- [4] Nori, H., King, N., McKinney, S. M., Carignan, D. & Horvitz, E. Capabilities of GPT-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375* (2023).
- [5] Singhal, K. *et al.* Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617* (2023). Published in Nature Medicine, 2024.
- [6] Asai, A. *et al.* Synthesizing scientific literature with retrieval-augmented language models. *Nature* (2026).
- [7] Taylor, R. *et al.* Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085* (2022).
- [8] Lewis, P. *et al.* *Retrieval-augmented generation for knowledge-intensive NLP tasks*, Vol. 33, 9459–9474 (2020).
- [9] Skarlinski, M. D. *et al.* Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740* (2024).
- [10] Asai, A., Wu, Z., Wang, Y., Sil, A. & Hajishirzi, H. *Self-RAG: Learning to retrieve, generate, and critique through self-reflection* (2024).

- [11] Edge, D. *et al.* From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).
- [12] Jiang, Z. *et al.* *Active retrieval augmented generation*, 7969–7992 (2023).
- [13] Kadavath, S., Conerly, T., Askell, A. *et al.* Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221* (2022).
- [14] Xiong, M. *et al.* *Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs* (2024).
- [15] de Moura, L. & Bjørner, N. *Z3: An efficient SMT solver*, 337–340 (Springer, 2008).
- [16] Neumann, M., King, D., Beltagy, I. & Ammar, W. *ScispaCy: Fast and robust models for biomedical natural language processing*, 319–327 (2019).
- [17] Sung, M. *et al.* BERN2: An advanced neural biomedical named entity recognition and normalization tool. *Bioinformatics* **38**, 4837–4839 (2022).
- [18] Orlando, R., Huguet Cabot, P.-L., Barba, E. & Navigli, R. *ReLiK: Retrieve and LinK, fast and accurate entity linking and relation extraction on an academic budget* (2024).
- [19] Bodenreider, O. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research* **32**, D267–D270 (2004).
- [20] Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports* **9**, 5233 (2019).
- [21] Gao, L., Ma, X., Lin, J. & Callan, J. *Precise zero-shot dense retrieval without relevance labels*, 1762–1777 (2023).
- [22] Wang, X. *et al.* *Self-consistency improves chain of thought reasoning in language models* (2023).
- [23] Es, S., James, J., Espinosa-Anke, L. & Schockaert, S. *RAGAS: Automated evaluation of retrieval augmented generation*, 150–163 (Association for Computational Linguistics, 2024).
- [24] Confident AI. DeepEval: The open-source LLM evaluation framework (2024). URL <https://github.com/confident-ai/deepeval>.
- [25] Khattab, O. *et al.* *DSPy: Compiling declarative language model calls into self-improving pipelines* (2024).
- [26] Liu, Y. *et al.* *G-Eval: NLG evaluation using GPT-4 with better human alignment*, 2511–2522 (Association for Computational Linguistics, 2023).

- [27] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q. & Artzi, Y. *BERTscore: Evaluating text generation with BERT* (2020).
- [28] Pal, A., Umapathi, L. K. & Sankarasubbu, M. Med-HALT: Medical domain hallucination test for large language models. *arXiv preprint arXiv:2307.15343* (2023).
- [29] Kim, S. *et al.* *Prometheus: Inducing fine-grained evaluation capability in language models* (2024).
- [30] Lee, J. *et al.* BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **36**, 1234–1240 (2020).
- [31] Luo, R. *et al.* BioGPT: Generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics* **23**, bbac409 (2022).
- [32] Singhal, K. *et al.* Large language models encode clinical knowledge. *Nature* **620**, 172–180 (2023).
- [33] Zakka, C. *et al.* Almanac—retrieval-augmented language models for clinical medicine. *NEJM AI* **1** (2024).
- [34] Zhao, X., Liu, S., Yang, S.-Y. & Miao, C. *MedRAG: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot* (2025).
- [35] Abo El-Enen, M., Saad, S. & Nazmy, T. A survey on retrieval-augmented generation (RAG) models for healthcare applications. *Neural Computing and Applications* **37**, 28191–28267 (2025).
- [36] Gao, Y. *et al.* Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2024).
- [37] Guu, K., Lee, K., Tung, Z., Pasupat, P. & Chang, M.-W. *REALM: Retrieval-augmented language model pre-training*, 3929–3938 (PMLR, 2020).
- [38] Karpukhin, V. *et al.* *Dense passage retrieval for open-domain question answering*, 6769–6781 (2020).
- [39] Khattab, O. & Zaharia, M. *ColBERT: Efficient and effective passage search via contextualized late interaction over BERT*, 39–48 (2020).
- [40] Borgeaud, S. *et al.* *Improving language models by retrieving from trillions of tokens*, 2206–2240 (PMLR, 2022).
- [41] Izacard, G. *et al.* Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research* **24**, 1–43 (2023).

- [42] Shi, W. *et al.* *REPLUG: Retrieval-augmented black-box language models*, 8364–8377 (2024).
- [43] Singh, A., Ehtesham, A., Kumar, S. & Khoei, T. T. Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv preprint arXiv:2501.09136* (2025).
- [44] Du, M. *et al.* A-RAG: Scaling agentic retrieval-augmented generation via hierarchical retrieval interfaces. *arXiv preprint arXiv:2602.03442* (2026).
- [45] Shi, Y. *et al.* *Reasoning in trees: Improving retrieval-augmented generation for multi-hop question answering* (2026). ArXiv:2601.11255.
- [46] Wei, J. *et al.* *Chain-of-thought prompting elicits reasoning in large language models*, Vol. 35, 24824–24837 (2022).
- [47] Yao, S. *et al.* *Tree of thoughts: Deliberate problem solving with large language models*, Vol. 36 (2023).
- [48] Besta, M. *et al.* *Graph of thoughts: Solving elaborate problems with large language models*, Vol. 38, 17682–17690 (2024).
- [49] Yao, S. *et al.* *React: Synergizing reasoning and acting in language models* (2023).
- [50] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K. & Yao, S. *Reflexion: Language agents with verbal reinforcement learning*, Vol. 36 (2023).
- [51] Wu, Q. *et al.* Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155* (2023).
- [52] Hong, S. *et al.* Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352* (2023).
- [53] Saxena, A., Tripathi, A. & Talukdar, P. *Improving multi-hop question answering over knowledge graphs using knowledge base embeddings*, 4498–4507 (2020).
- [54] Su, X. *et al.* KGAREvision: An AI agent for knowledge-intensive biomedical QA. *arXiv preprint arXiv:2410.04660* (2024).
- [55] Chandak, P., Huang, K. & Zitnik, M. Building a knowledge graph to enable precision medicine. *Scientific Data* **10**, 67 (2023).
- [56] Peng, B. *et al.* Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921* (2024).
- [57] Himmelstein, D. S. *et al.* Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife* **6**, e26726 (2017).

- [58] Cheng, H. *et al.* Scaling biomedical knowledge graph retrieval for interpretable reasoning: Applications to clinical diagnosis prediction. *medRxiv* (2026).
- [59] Guo, J., Li, J., Li, Y., Xie, P. *et al.* *A survey of confidence estimation and calibration in large language models* (2024).
- [60] Taubenfeld, A. *et al.* *Confidence improves self-consistency in LLMs* (2025).
- [61] Quach, V. *et al.* *Conformal language modeling* (2024).
- [62] Chen, Y. *et al.* The fusion of large language models and formal methods for trustworthy AI agents: A roadmap. *arXiv preprint arXiv:2412.06512* (2024).
- [63] Ren, Z. *et al.* Towards reliable medical LLMs: Benchmarking and enhancing confidence estimation of large language models in medical consultation. *arXiv preprint arXiv:2601.15645* (2026).
- [64] Liu, X. *et al.* *Uncertainty quantification and confidence calibration in large language models: A survey* (2025).