Knowledge Graph Research Report

Foundations design computation reasoning and agents

知识图谱研究报告

An illustrated English report with Chinese annotations for key concepts and technical terms.

This report explains how knowledge graphs are designed, constructed, stored, analyzed, and used. It connects the research literature to an original evidence graph case study and examines how language models and agents can use graph structure while preserving the status and origin of their conclusions.

14 chapters · 14 original figures · 52 references

Approximately 20,000 English words, with a bilingual glossary and reproducibility notes.

1 October 2026

Literature discovery with Consensus and SciSpace
Numerical verification with Wolfram
Original synthesis and illustrative system designs

Contents

Executive abstract 3
1 Meaning and foundations of knowledge graphs 4
2 Schema design and semantic commitments 6
3 Entity extraction and identity resolution 10
4 Relation extraction and evidence preservation 13
5 Construction pipelines and knowledge refinement 17
6 Knowledge storage and query architecture 19
7 Knowledge computation and graph algorithms 23
8 Knowledge reasoning and the status of conclusions 26
9 Applications and the conditions for value 30
10 Language models retrieval and GraphRAG 33
11 Agents and knowledge graphs 36
12 Evaluation quality and governance 39
13 A complete Research Evidence Graph implementation 43
14 Research priorities and conclusions 46
Appendix A Reproducibility and interpretation notes 49
Appendix B English and Chinese terminology 51
References 53

Reading guide

Chapters 1 and 2 explain meaning and schema. Chapters 3 through 6 cover construction and storage. Chapters 7 and 8 separate computation from reasoning. Chapters 9 through 11 examine applications, language models, and agents. Chapters 12 through 14 address evaluation, implementation, and research priorities. English terms receive Chinese annotations on first use and in the glossary. Figures are original explanatory diagrams; numeric examples are synthetic calculations rather than empirical findings. Square bracket numbers point to the reference list.

Executive abstract

Knowledge graphs（知识图谱）make the identities, relationships, and context of information explicit. Their practical value comes from connecting records that would otherwise remain separated, expressing what those connections mean, and preserving the evidence needed to interpret them. A graph becomes useful knowledge infrastructure when people can determine which entity a statement concerns, where the statement came from, when it applies, and what conclusions can legitimately follow. A visually impressive network alone does not establish any of these properties.

This report examines knowledge graphs as an integrated research and engineering subject. It connects schema design（模式设计）, entity extraction（实体抽取）, relation extraction（关系抽取）, storage（知识存储）, graph computation（图计算）, reasoning（知识推理）, applications, and artificial intelligence agents（智能体）. The central argument is that reliable performance depends on agreements between these stages. An extraction model cannot repair an ambiguous identity policy. A fast database cannot make an unsupported relationship true. An agent cannot safely act on a graph whose permissions, timestamps, or evidential status are missing.

Three distinctions organize the analysis. First, a mention in a document differs from the entity it denotes. Second, a source assertion differs from an accepted fact and from a model prediction. Third, a graph algorithm, a logical proof, and a causal conclusion answer different questions. Keeping these distinctions visible improves system design and prevents misleading claims about what a knowledge graph understands. They also explain why storing everything as an undifferentiated collection of triples often produces disappointing results.

The report combines a narrative literature review with original architectural analysis and synthetic worked examples. Foundational surveys provide orientation; standards establish formal semantics; primary research papers illustrate specific methods. Consensus and SciSpace were used for literature discovery, followed by checks against original publication records and accessible paper pages. Wolfram was used to verify a small PageRank calculation. Search indexes sometimes disagree about publication years or author names, so final references favor the identifiable published version where available. The evidence search was conducted on 1 October 2026. This is a selective integrative review, not an exhaustive systematic review or a new benchmark study.

Recent work on graph retrieval augmented generation and graph guided agents shows several productive ways to combine structured knowledge with language models. The evidence supports investigating these combinations for questions that require explicit connections, provenance, or controlled updates. It does not justify assuming that a graph always improves a retrieval system. Construction errors, retrieval expansion, stale summaries, and additional processing costs can offset the benefits. A strong evaluation therefore compares graph based methods with simpler alternatives under the same corpus, model, access policy, and cost budget.

An original example called the Research Evidence Graph runs through the report. It links documents, claims, people, methods, datasets, and experiments. All people, organizations, dates, and experimental records in this example are fictional unless a passage explicitly identifies a real publication. The example demonstrates how a graph can support a research assistant without confusing an author's claim with independent confirmation. It also provides a practical route from a small local project to a governed organizational system.

1 Meaning and foundations of knowledge graphs

1 1 An operational definition

A knowledge graph is a structured representation of entities（实体）and the relationships（关系）among them, organized so that the representation can support interpretation, integration, querying, and inference. This report uses that operational definition rather than treating one storage format as the universal definition. A useful graph may describe people and institutions, components and dependencies, genes and biological processes, or documents and claims. Its subject matter can be concrete, abstract, changing, disputed, or only partially observed. The important requirement is that the graph communicate enough meaning for its intended users to interpret the connections consistently.

Hogan and colleagues survey graph data models, querying, validation, representation, and reasoning as connected aspects of knowledge graphs [1]. Ji and colleagues organize research around representation learning, acquisition and completion, temporal knowledge, and applications [2]. These surveys illustrate the field's breadth. Some research communities emphasize formal semantics and interoperability; others emphasize learned representations, large scale information extraction, or application performance. A report on the subject must make its assumptions explicit because the same phrase can refer to systems with very different guarantees.

Consider three records: a document lists “M. Chen” as an author, an institutional directory lists “Mei Chen,” and an experiment log identifies “researcher 042.” A knowledge graph can link these records to a shared person if sufficient evidence supports that identification. It can then connect the person to publications and experiments. The benefit is not that the information is drawn as circles and lines. The benefit is that the system can ask a question across the records while retaining the grounds for believing they concern the same person.

This definition does not require a graph to contain all human knowledge, to be automatically generated, or to use a specific database. A carefully maintained graph of several hundred research claims can be more useful for a focused task than a much larger collection of weakly identified entities. Scale changes engineering requirements, but it does not substitute for semantic clarity. Nor does a graph need every possible ontology feature. A modest vocabulary with consistent identity and provenance may provide a strong foundation for a practical application.

1 2 Entities relations and statements

An entity is something the system chooses to identify persistently. A node（节点）is a representation of an entity or another modeled object. These terms are related but not interchangeable: one entity may be represented by multiple unresolved nodes, while a node might represent a document fragment, an assertion, or an event rather than a physical object. A relation type states the intended meaning of a connection. A particular edge（边）instantiates such a connection, potentially with additional attributes.

The triple（主语谓语宾语三元组）is a familiar statement form: subject, predicate, object. For example, “Paper P uses Method M” identifies a subject, names a relationship, and identifies an object. In RDF, a graph is a set of triples; subjects are IRIs or blank nodes, predicates are IRIs, and objects may also be literals. Datasets can organize a default graph and named graphs [4]. This formal model is precise, but a real claim

often needs additional context. “Method M improves accuracy” is incomplete without an evaluation, a baseline, a dataset, and a measure.

The Research Evidence Graph therefore represents a comparative result as an Evaluation entity. That entity connects a method, dataset version, metric, baseline, score, and source passage. This choice prevents two evaluations from being accidentally collapsed into a single timeless edge. It also exposes what is missing. If the source provides no baseline, the record can preserve that absence instead of implying that the method improved on every alternative. The graph becomes a structured account of evidence rather than a collection of disconnected slogans.

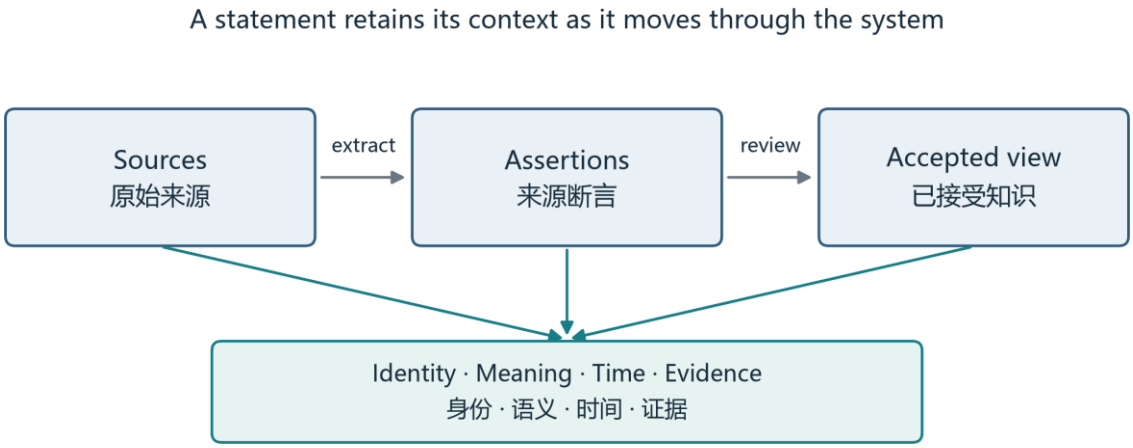


Figure 1. Knowledge representation retains identity, meaning, time, and evidence across source interpretation and acceptance. Original conceptual diagram.

1 3 Schema and instance knowledge

The schema（模式）describes available categories, relationship meanings, and constraints. Instance data（实例数据）describes particular things. A statement that every experiment has a dataset is a schema level commitment; a statement that Experiment E uses Dataset D is an instance assertion. The distinction allows a system to check whether particular records satisfy expectations. It also makes model changes visible. Adding a new dataset record usually does not require changing the schema, while introducing a new concept of dataset lineage might.

An ontology（本体）typically makes conceptual commitments more explicit than a lightweight application schema. It may define class hierarchies, property characteristics, equivalence, disjointness, or logical restrictions. A taxonomy（分类体系）primarily organizes categories hierarchically. A controlled vocabulary（受控词表）standardizes labels. These artifacts can coexist. For example, a research graph might use a controlled vocabulary for research methods, a taxonomy for scientific disciplines, and a small ontology for authorship and evidence. Calling every list of labels an ontology obscures these different functions.

A graph database is a storage and query system. A knowledge graph is the modeled body of knowledge and its associated interpretation. The same knowledge graph can be stored across several physical systems, and a graph database can contain information with little explicit semantics. Similarly, a relational database can

support parts of a knowledge graph through tables, identifiers, and joins. Choosing a graph database is therefore an implementation decision; it does not by itself settle the conceptual design.

1 4 Knowledge is situated and incomplete

An assertion（断言）records that a statement has been made. Its existence does not establish its truth. A source may be mistaken, outdated, promotional, or describing a hypothetical situation. The graph should distinguish “Document D reports claim C” from “Claim C has been independently verified.” This distinction is especially important when language models extract confident prose. The surface certainty of a sentence and the reliability of the underlying evidence are separate properties.

Missing information also requires interpretation. Under an open world assumption（开放世界假设）, the absence of a statement does not imply that it is false. RDF and OWL reasoning commonly operate with open world semantics [5,7]. A local application can still define a closed and audited dataset for a specific purpose. For example, an experiment registry might promise that every completed experiment has been registered by the end of a reporting period. A query about that registry then has a narrower completeness assumption, which should be documented rather than silently generalized to the entire world.

Contradiction can be informative. Two papers may report different outcomes because they use different populations, dates, instruments, or definitions. An integration pipeline should first test whether the claims are actually about the same conditions. If disagreement remains, representing both claims with provenance can be more faithful than choosing one automatically. The graph can support investigation by exposing the point of conflict. A single resolved value should be treated as a governed view with a stated resolution policy, not as an inevitable consequence of collecting data.

1 5 When a graph is useful

A graph is particularly useful when questions repeatedly cross identities, relationship types, sources, or levels of abstraction. Examples include tracing a component failure through supplier dependencies, finding research claims supported by several independent studies, and identifying which internal procedures depend on an outdated regulation. In each case, explicit links allow the application to follow connections that are difficult to reconstruct reliably from isolated text fragments.

The investment is less compelling when the workload consists mainly of retrieving complete documents by keyword, calculating regular numeric aggregates, or maintaining a stable transactional form with few cross record relationships. Those tasks may be served well by search engines, analytical databases, or conventional application tables. A graph can coexist with those systems, but its additional schema, identity, and maintenance work should earn its place through measurable task improvements. This report treats that decision as an empirical design question rather than a commitment to one technology.

2 Schema design and semantic commitments

2 1 Begin with questions and decisions

Schema design starts with the questions a system must answer and the decisions those answers support. A competency question（能力问题）is a concrete test of what the model should express. For the Research Evidence Graph, one question is: “Which studies support the claim that Method M improves retrieval quality

on Dataset D, and under what conditions?" Another is: "Which conclusions depend on a paper that has been corrected?" These questions require sources, claims, evaluations, versions, and dependency links. They immediately reveal why a graph containing only Paper, Author, and Topic nodes would be insufficient.

A useful design workshop takes several real questions and writes the evidence path needed for each answer. The team then identifies the minimum concepts required to express those paths. This process avoids two common errors: importing an enormous ontology before understanding the task, and drawing a convenient schema that cannot answer the important questions. Questions should include exceptions. The model must express a withdrawn claim, an unknown author identity, an experiment with multiple datasets, and a result whose numerical score is unavailable.

Different stakeholders may need different views of the same graph. A librarian cares about bibliographic identity; a researcher cares about experimental comparability; an administrator cares about access and ownership. The shared core should contain stable concepts on which they can agree. Specialized views can extend that core without forcing every user to adopt every detail. A schema is most maintainable when it separates common identity commitments from application specific presentation choices.

2 2 Choose the correct unit of representation

Modeling granularity（建模粒度）determines what receives an identity. A research project, a paper, a particular edition of that paper, and a paragraph in that edition are different objects. If a correction changes a paragraph, a graph that identifies only the paper cannot precisely locate the change. If every character receives a permanent node, maintenance may become unnecessarily expensive. The appropriate granularity follows the retrieval and audit requirements. A paragraph can often be stored as a source anchored span without becoming a heavily connected domain entity.

A useful rule is to create a node when an object needs independent identity, relationships, lifecycle, or provenance. Use a property（属性）when a value is naturally attached to one object and does not need independent connections. A publication date can usually be a property. An organization should usually be an entity because authors, projects, contracts, and events may refer to it. A research method may initially be a text property, but repeated questions about versions, families, and evaluations can justify promoting it to a node.

Events（事件）and qualified relations deserve particular care. "Person A worked for Organization B" is a relation that may change over time. Representing an Employment event or association with start and end dates can preserve multiple appointments and conflicting records. In the research example, an Evaluation plays a similar role. It connects several participants and conditions that belong together. A binary edge with many attributes may suffice in a property graph, while an explicit event node can make those attributes independently linkable and easier to exchange across models.

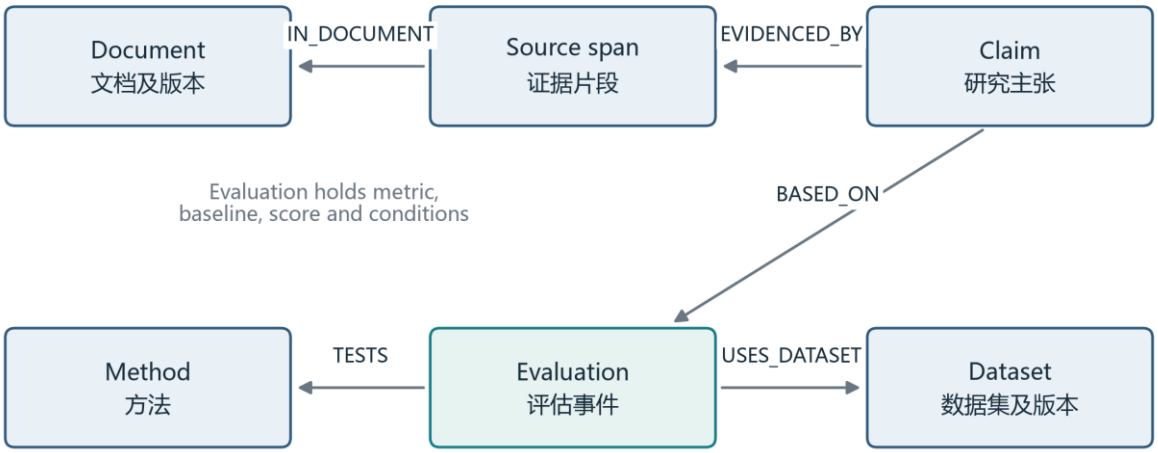


Figure 2. A minimum evidence schema. An Evaluation connects methods and dataset versions; Claims retain source evidence. Relationship names illustrate the proposed design.

2 3 Identity before labels

An identifier（标识符）should remain stable when a display name changes. A paper title is a poor primary key because titles can contain punctuation differences, corrections, translations, or duplicate wording. A DOI is useful when it exists and has been verified, but a system also needs a local identifier for documents without one. Local identifiers should not pretend to be global identifiers. Keeping the namespace explicit makes later reconciliation easier.

Identity policies must specify both merge and split behavior. A merge means that records previously treated separately are now considered to denote the same entity. A split reverses an erroneous merge. If downstream edges are copied without recording their original source, splitting becomes difficult. The Research Evidence Graph keeps mention level provenance and a versioned mapping from mentions to canonical entities. This allows a correction to revise identity links without deleting the underlying documents or losing the history of the mistake.

Names should support multilingual use. An English preferred label, a Chinese preferred label, abbreviations, and historical aliases can all refer to one concept. SKOS provides a standardized vocabulary for concept schemes and labels, including broader and narrower relationships [11]. A practical application can reuse these distinctions even when its storage is a property graph. It should still decide whether two labels are translations of one concept or names for related but distinct concepts. Translation similarity alone does not establish conceptual equivalence.

2 4 Formal semantics and operational validation

RDF Schema offers classes, subclass relationships, and domain and range descriptions [6]. A subtle point is that domain and range declarations support inference about types. They are not equivalent to a database check that rejects every unexpected value. If a property has Person as its domain and a statement uses that property with an unknown subject, an RDFS reasoner can infer that the subject is a Person. A team expecting automatic rejection may therefore obtain behavior different from its operational intention.

OWL adds constructs for richer logical descriptions, including class restrictions and property characteristics. It does not assume that differently named individuals are necessarily different, and many conclusions depend on whether explicit inequality has been asserted [7]. For implementation planning, OWL profiles provide restricted language fragments aimed at different reasoning needs and computational tradeoffs [12]. Choosing an expressive language without a corresponding reasoning strategy can leave a project with declarations that no deployed component actually evaluates.

SHACL（形状约束语言）describes constraints for validating RDF data graphs [8]. An application could require each accepted Evaluation to have a dataset reference and a numeric score with an appropriate datatype. A validation report then identifies records that fail the defined shapes. Such validation concerns conformance to a declared contract. It does not establish that a reported score was measured honestly or that the experiment was well designed. The Research Evidence Graph therefore records validation status separately from scientific confidence.

In a property graph implementation, uniqueness constraints can prevent duplicate identifiers, while application logic can check more complex requirements. Neo4j documents several constraint types with version and edition conditions [45]. A portable design should specify the required invariant first and then identify where it is enforced: database, ingestion service, batch audit, or human review. This prevents the schema diagram from implying protections that exist only as comments in a document.

2 5 Time uncertainty and provenance

Temporal modeling distinguishes valid time（有效时间）, when a statement applies in the modeled world, from transaction time（记录时间）, when the system recorded it. Suppose a correction published in March states that an affiliation ended in January. The affiliation's valid interval changes in January, but the graph learned about the change in March. Keeping both times allows a user to ask what was believed on a past date and what is currently believed about that same past period.

An uncertain claim needs more than a single confidence number. Extraction confidence estimates whether the source was interpreted correctly. Source reliability concerns the origin of the statement. Empirical uncertainty concerns the measurement or statistical result. A fourth dimension may concern whether the statement transfers to a new population. Combining these into one unlabeled score destroys useful information. A graph can store separate fields and require an application to explain which dimensions affect its decision.

Provenance（溯源）links records to sources, activities, and responsible parties. PROV-O offers a vocabulary for entities, activities, and agents and their relationships [10]. The report's proposed implementation records document version, evidence span, extraction run, model version, reviewer, and acceptance decision. This operational extension is a design recommendation rather than a claim that every field is mandated by PROV-O. The aim is to make a statement reproducible and correctable at the level where errors actually occur.

2 6 Schema evolution and governance

A schema changes when users discover a distinction that matters. Initially, a project might treat all relationships between papers as CITES. Later it may need SUPPORTS, CRITIQUES, REPLICATES, and USES_DATA_FROM. Replacing every old citation with one of these finer categories would invent knowledge unless the sources are reexamined. A better migration retains the broad relation, adds specialized relations when supported, and marks which records have been reviewed under the new schema.

Versioning should cover the schema, mapping rules, validation shapes, and query templates. A migration plan explains how existing records remain interpretable and how dependent applications are tested. Deprecating a relation should not silently change its historical meaning. The team can introduce a new relation, maintain a compatibility view for a defined period, and document the semantic difference. These measures protect users who rely on old exports or cached analysis results.

Schema governance is a decision process, not only a technical repository. A domain specialist should resolve whether two concepts are meaningfully distinct. A data steward should own identity and provenance policies. An engineer should verify enforcement and migration feasibility. For a small personal research graph, one person may fill all three roles, but the questions remain distinct. Recording the rationale for a modeling choice helps that person avoid repeating the same debate months later.

The resulting schema should be small enough to explain and rich enough to answer the competency questions. Its quality can be tested through representative records, ambiguous cases, and adversarial examples. A useful review asks whether each node type has a clear identity policy, each relationship has a direction and meaning, and each important constraint has an enforcement location. Once those commitments are stable, extraction and storage become much easier to evaluate.

3 Entity extraction and identity resolution

3 1 From text spans to canonical entities

Entity extraction is a sequence of related tasks. Mention detection（提及检测）finds a span of text. Named entity recognition（命名实体识别）assigns a category to that span. Entity linking（实体链接）maps it to an existing entity when appropriate. Coreference resolution（共指消解）connects expressions that refer to the same thing in context. Entity resolution（实体消歧与归一）reconciles records across sources. A system can perform well on mention boundaries while still making serious identity errors.

Consider the fictional passage: “Mei Chen presented GraphLens at Northbridge Institute. Chen later released its evaluation dataset.” The first sentence introduces a person, a method or software system, and an institution. The second sentence requires deciding that “Chen” refers to the same person and determining what “its” refers to. A pipeline should preserve the original spans and the alternative interpretations. If GraphLens is also the name of an unrelated commercial product, linking by surface name alone would create a misleading connection.

A canonical entity（规范实体）is the system's current representation of a referent. It should have a stable identifier and a record of the evidence used to attach mentions. New mentions should not automatically create new canonical entities. Conversely, unknown entities should not be forced into the closest existing candidate. A NIL or unresolved outcome is often the correct result. The graph can later revisit unresolved mentions when more information becomes available.

3 2 Methods and their appropriate roles

Dictionary and pattern based extraction remain useful when identifiers follow strict formats or a vocabulary is small and stable. A DOI pattern, an equipment serial number, or a curated list of internal project codes may be easier to extract with deterministic rules than with a large model. Rules are also transparent: a reviewer

can inspect the exact matching condition. Their weakness is incomplete coverage and fragility when text varies. Combining rules with learned models can preserve precision for standardized fields while improving recall for natural language expressions.

Lample and colleagues demonstrated neural approaches to named entity recognition using character and word representations with sequence modeling [13]. BERT introduced contextual language representations that became a foundation for many extraction systems [14]. These contributions concern representation and prediction; they do not remove the need for a domain appropriate annotation policy. A model trained to recognize Person, Organization, and Location will not automatically understand the distinction between a scientific method, a metric, and a dataset version.

Span based approaches represent candidate text spans and classify them, making it easier to express overlapping or nested mentions. SpERT combines transformer representations with entity and relation extraction [16]. DyGIE++ uses contextualized span representations across entity, relation, and event extraction tasks [17]. These examples illustrate alternatives to a simple token tagging pipeline. In practice, the choice depends on whether the corpus contains nested terminology, cross sentence references, or multiple relations involving the same expression.

Large language models can produce structured entity candidates from instructions and examples. Their flexibility is valuable when a domain lacks a large annotated dataset. A production extractor should nevertheless constrain the output schema, require source spans, validate identifiers and types, and provide an abstention route. A plausible generated entity that never appears in or follows from the source should remain outside the accepted graph. Asking the model to be accurate is weaker than checking the observable properties of its output.

Table 1. Representative extraction approaches

Approach	Useful capability	Evaluation concern
Rules and dictionaries	Precise standardized identifiers	Coverage and format variation
Sequence tagging [13,14]	Contextual mention and type prediction	Nested mentions and domain transfer
Span and joint models [16,17]	Shared entity and relation representations	Propagation of argument errors
Generative extraction [19]	Structured outputs from text generation	Unsupported additions and missing qualifiers
Entity retrieval and ranking [15]	Scalable candidate resolution	NIL handling and incorrect merges

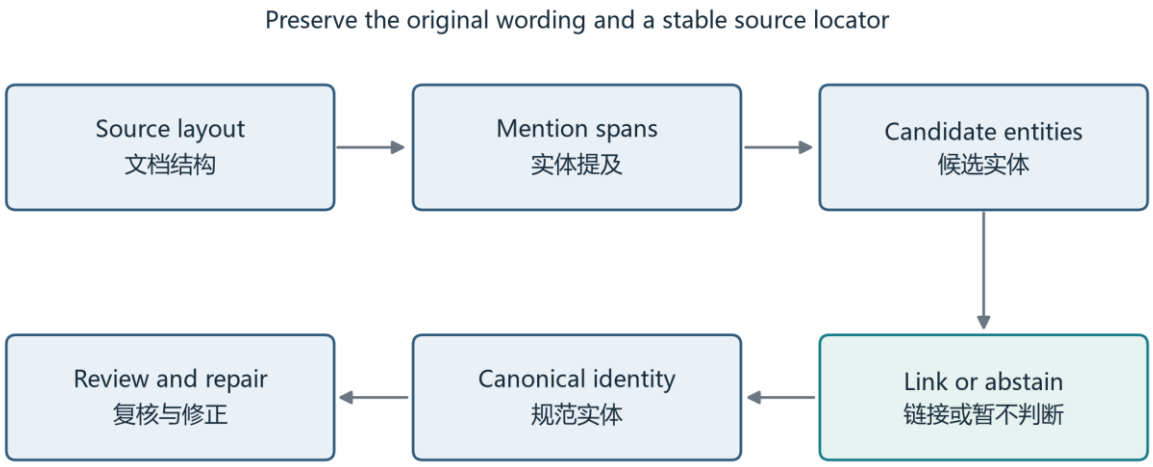


Figure 3. Entity extraction separates mention detection, candidate generation, linking, and review. Abstention preserves uncertainty instead of forcing an identity.

3 3 Linking and resolution

Entity linking usually begins with candidate generation（候选生成）and proceeds to candidate ranking（候选排序）. Candidate generation should favor recall: the correct entity must be present before a ranker can select it. A ranker then compares context, aliases, descriptions, and available identifiers. BLINK uses dense retrieval followed by more detailed candidate scoring to support scalable entity linking [15]. The practical lesson is architectural: an inexpensive retrieval stage and a more selective disambiguation stage can serve different purposes.

The Research Evidence Graph uses exact identifiers first, then normalized titles and context, then a review queue for uncertain matches. An exact DOI match is strong evidence for document identity after validation. A similar title plus a different year and author list requires investigation. For people, affiliation and coauthor context may help, but they should not be treated as permanent identity attributes. A person can move institutions, and two people can share both a name and an organization.

Resolution decisions have asymmetric costs. Failing to merge two records may duplicate information, while incorrectly merging two researchers can attribute publications and claims to the wrong person. The acceptable threshold therefore depends on the downstream task. A discovery interface may show possible matches. A system used for attribution or evaluation should require stronger evidence before merging. The threshold should be chosen using labeled examples and review costs, not merely because a model emits a convenient decimal score.

An audit trail should preserve merge justifications and allow reversal. One practical representation stores a canonical entity separately from its source records and attaches a status to each mapping. If a merge is later rejected, the system changes the mappings and recomputes derived views. This is safer than destructively replacing every original identifier. The same design makes it possible to explain why an entity has a particular label or attribute.

3 4 Language layout and domain difficulties

Chinese and English mixed documents create particular extraction challenges. Chinese word boundaries are not always marked by spaces, abbreviations may remain in English, and translated names can vary across sources. Normalization should preserve the original surface form alongside any standardized form. A multilingual alias table can help retrieval, but aliases should include language and source information. Otherwise, an abbreviation that is unambiguous in one domain may incorrectly merge entities in another.

Documents also contain meaning in layout. A dataset name in a table header may apply to several rows; a footnote may qualify a numerical value; a figure caption may define an abbreviation used nowhere else. Flattening a PDF into a single text stream can destroy these associations. A reliable pipeline captures sections, tables, captions, and reading order before extraction. For scanned material, optical character recognition（光学字符识别）errors should be distinguished from the extraction model's errors because they require different remedies.

Specialized domains require careful type definitions. “Transformer” may name a model family, a particular architecture, or an electrical device. “Control” may be a study group, a procedure, or an instruction. The correct label depends on context and the graph's purpose. Annotation guidelines should include positive examples, near misses, and boundary cases. Disagreements between annotators can reveal an unclear schema rather than merely careless labeling.

3 5 Evaluation and review

Entity extraction evaluation should separate span accuracy, type accuracy, linking accuracy, and end to end canonicalization. Exact span matching is useful for reproducibility, while relaxed overlap measures can diagnose near misses. Reporting only a combined score hides whether errors originate in detection or identity. Evaluation data should contain realistic documents, including low quality scans, multilingual passages, unseen names, and absent entities. An easy benchmark with familiar names may substantially overstate deployment quality.

Review can be prioritized by consequence and uncertainty. A low confidence mention with no downstream use may wait; an uncertain identity supporting many important claims should be reviewed early. The queue can also sample high confidence predictions to detect systematic errors that confidence based selection would miss. Reviewer corrections should update the labeled evaluation set only through a controlled process, with a separate holdout set retained for measuring improvement.

The output of entity extraction should therefore be a set of evidence anchored candidates with identity decisions and recorded uncertainty. Treating extraction as a one step conversion from text to truth makes later repair expensive. Treating it as a sequence of reversible interpretations allows the knowledge graph to improve as sources, models, and human understanding change.

4 Relation extraction and evidence preservation

4 1 Define what a relation asserts

Relation extraction asks whether a source expresses a particular relationship between identified arguments. The arguments may be entities, events, quantities, or claims. The task differs from simply finding two terms near each other. “GraphLens was compared with PathSearch” supports a comparison relation. It does not

establish that either method outperformed the other. “The authors hope GraphLens will improve recall” expresses an intention or hypothesis, not an observed result. A graph construction system must preserve these distinctions if the graph will later support evidence based answers.

A relation inventory（关系集合）should define argument types, direction, meaning, and evidence requirements. USES_DATASET links an experiment to a dataset; AUTHORED_BY links a document to a person; REPORTS links a document to a claim. SUPPORTS and CONTRADICTS require more careful interpretation because support is usually conditional. A study can support one aspect of a claim while challenging another. The schema should either represent the relevant claim precisely or use a qualified assessment object that identifies the dimension being assessed.

Relationship direction is not a cosmetic choice. If a paper cites another paper, reversing the edge changes the interpretation of citation counts and traversal. If a company supplies another company, reversing the edge changes supply risk analysis. Extraction guidelines should include inverse formulations: “Dataset D was used by Experiment E” and “Experiment E used Dataset D” describe the same directed relation after normalization. Direction errors are easy to miss in a graph visualization because the connected entities still look plausible.

4 2 Sentence and document level extraction

A sentence level classifier predicts a relation for a pair of mentions within a sentence. It can exploit surrounding words and grammatical structure. This setting is useful when the relevant evidence is local and the schema is narrow. Its limitation appears when an argument is introduced elsewhere, a pronoun replaces an entity, or the relation depends on a sequence of statements. Treating every sentence independently then loses the information required for a correct interpretation.

DocRED was developed to support document level relation extraction, including cases requiring information across sentences [18]. The general lesson is that the unit of inference can be larger than the unit of mention detection. In the fictional example, one paragraph identifies GraphLens as the proposed method, another introduces Dataset D, and a results table reports the score. The accepted Evaluation record may therefore need evidence from several locations. A single sentence pointer would give a misleading impression of evidential completeness.

Joint extraction（联合抽取）models can share information between entity recognition and relation prediction. A likely USES_DATASET relation can help distinguish a dataset name from a method name, while entity types can constrain plausible relation labels. SpERT and DyGIE++ illustrate approaches that connect these tasks through contextual representations [16,17]. This coupling can improve consistency, but it can also spread an early mistake. Evaluation should therefore inspect both the final relation and the argument identities on which it depends.

Generative extraction（生成式抽取）expresses relational outputs as a sequence. REBEL provides a notable example of relation extraction framed as language generation [19]. Such methods can produce several triples from a passage without a separate classifier for every candidate pair. The operational challenge is to distinguish valid structured output from faithful extraction. A syntactically correct triple may still add a relation absent from the source, omit a qualifier, or use a noncanonical entity label. Parsing is only the beginning of validation.

4 3 Negation modality and scope

Negation（否定）changes the meaning of a statement, but its scope can be difficult to determine. “The study did not find evidence that M improves recall” is not equivalent to “M reduces recall.” It also differs from “M never improves recall.” A graph that stores only a positive or negative edge loses these distinctions. The Research Evidence Graph represents the reported claim, its polarity, its modality, and the evaluation context. The accepted wording remains close enough to the source that a reviewer can check the interpretation.

Modality（情态）covers possibility, necessity, intention, expectation, and conditionality. “M may reduce cost” describes a possibility; “M must be run with parameter P” describes a requirement; “If the dataset is balanced, M improves precision” is conditional. An extractor that drops modal verbs can turn tentative language into a strong factual assertion. A practical test set should deliberately include these constructions rather than rewarding only simple affirmative sentences.

Quantifiers（量词）also matter. “Some studies report an improvement” and “All studies report an improvement” cannot be stored as the same relation. The first may be supported by a single study; the second depends on the defined population and completeness of the search. A claim about a proportion needs a denominator. If the source does not specify the population, the graph should preserve the phrase as a claim with unresolved scope rather than inventing a universal statement.

Temporal scope can create apparent contradictions. “Institute N hosts Project P” in an old annual report may coexist with a later report naming a different host. The extraction stage should capture dates and source context before a reconciliation policy chooses a current view. Otherwise, a later merge may erase the historical relationship or incorrectly classify one report as false. Both historical and current questions benefit from retaining the original statements.

4 4 Supervision and annotation strategies

Supervised extraction requires labeled examples that reflect the target domain. A useful annotation record identifies arguments, relation type, evidence span, qualifiers, and uncertainty. The annotation policy should distinguish “not expressed” from “expressed negatively” and from “cannot decide.” These outcomes may eventually be collapsed for a particular model, but preserving them in the reference data supports better error analysis. It also helps explain why human annotators disagree.

Distant supervision（远程监督）uses existing database relations to label text automatically. Mintz and colleagues developed an influential approach based on aligning known relations with textual occurrences [20]. The attraction is scale: existing structured knowledge can generate many training examples. The assumption is imperfect because a sentence mentioning both entities may discuss a different relationship. In an applied pipeline, such labels should be treated as noisy evidence for training rather than as verified gold annotations.

Weak supervision（弱监督）can combine rules, dictionaries, existing databases, and model predictions. A sensible project begins with high precision patterns for well defined relations, then identifies what those patterns miss. Human review can focus on examples where labeling sources disagree or where new relation types appear. This design makes annotation effort responsive to the actual failure modes of the system. It also avoids assuming that a large automatically labeled dataset necessarily provides broader conceptual coverage.

Active learning（主动学习）selects examples expected to improve a model or resolve consequential uncertainty. In the Research Evidence Graph, examples involving comparison language and multiple baselines might be especially informative because those errors affect the central application. Selection should not rely only on uncertainty scores, which can overlook confidently wrong patterns. A mixed sample containing uncertain, diverse, and randomly selected records gives a more informative review set.

4 5 A proposed acceptance contract

Before a candidate relation becomes an accepted graph assertion, the proposed system checks that its arguments exist, their types are compatible, the relation belongs to the schema, and the source locator resolves. It then checks whether the extracted wording preserves negation, time, and essential qualifiers. These checks do not all need one model. Deterministic validation can handle identifiers and datatypes, while a separate model or human reviewer can assess semantic faithfulness. The result records which checks ran and which remained unresolved.

For a comparative result, the contract is deliberately stricter. An accepted record needs the compared methods, dataset version, metric definition, and the reported outcome. If a source states only that a method is “better,” the system can store that as an attributed qualitative claim, but it should not manufacture a numeric improvement. If a table reports several metrics, the graph creates distinct metric observations tied to the same Evaluation rather than selecting the most favorable one.

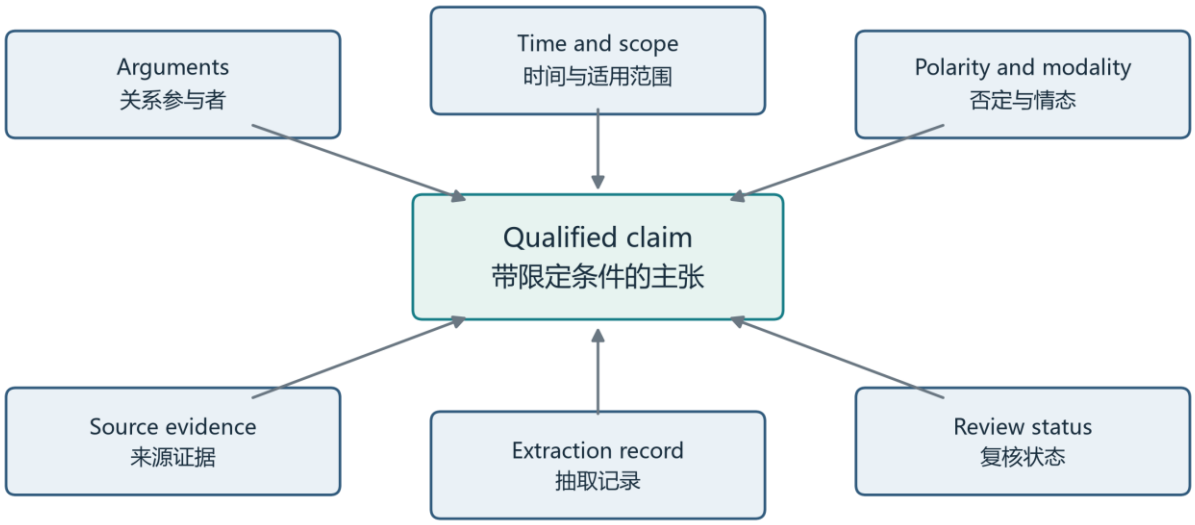


Figure 4. A claim requires context beyond its subject and object. The diagram summarizes the report’s proposed acceptance record.

The graph should preserve the difference between extraction and corroboration. Two documents can repeat the same unsupported assertion. Counting both as independent confirmation would exaggerate the evidence. Citation links, shared datasets, copied text, and common authorship can help identify dependence, but none alone provides a perfect independence test. The interface should therefore expose the evidence chain and allow a reviewer to examine whether apparently separate sources originate from the same study.

4 6 Measure useful extraction quality

Relation precision asks how many predicted relations are correct under the annotation policy. Recall asks how many reference relations were recovered. End to end scoring should require the correct argument identities, direction, relation label, and any qualifiers essential to the task. A separate diagnostic score can relax one requirement at a time. This helps distinguish a model that understands the relationship but misses an entity boundary from one that confidently reverses the result of an experiment.

Evidence quality deserves its own measure. A relation may happen to be true even though the cited passage does not support it. A research assistant must be evaluated on whether its evidence actually entails or justifies the reported statement. Human reviewers can label evidence sufficiency and identify missing context. For deployment, a smaller set of well supported relations can be more valuable than a large graph with impressive recall and weak evidential traceability.

5 Construction pipelines and knowledge refinement

5 1 Treat construction as an iterative process

Knowledge graph construction is usually an ongoing process of acquisition, reconciliation, validation, and revision. Zhong and colleagues review acquisition, refinement, and evolution as major parts of automatic construction [3]. This broader view is useful because a graph rarely stays correct after a single import. Sources change, extractors improve, and users discover mistakes. A design that anticipates revision can turn those changes into routine maintenance rather than repeated reconstruction from scratch.

The proposed pipeline begins with a source inventory. Each source receives an identifier, owner, access policy, format, update schedule, and permitted use. A content hash identifies an exact file version, while metadata records the acquisition date and original location. The raw source is retained separately from extracted text. This separation lets the team diagnose whether an error came from the original document, text conversion, model interpretation, or later reconciliation.

Parsing produces structural units such as sections, paragraphs, tables, captions, and references. Chunking（分块）then creates processing units while preserving links back to the document structure. Chunk size should follow the task. Entity detection may need local context, while interpreting a result can require the methods section and table notes. Overlapping chunks help preserve context but create duplicate candidates, so the merge policy must recognize that several extractions may come from the same underlying passage.

5 2 Separate staging from accepted knowledge

A staging area（暂存区）holds candidate entities and claims before acceptance. The main graph serves records that satisfy the current acceptance policy. This separation gives the system a place to keep uncertain outputs without presenting them as established knowledge. Rejected candidates can remain available for debugging with appropriate retention controls. The staging area should record the reason for rejection so that the team can distinguish malformed output from unsupported meaning.

An extraction run receives a versioned configuration: model, prompt, schema version, parser, chunker, and relevant parameters. Reprocessing the same document with a new model produces a new run rather than silently replacing the old record. The system can compare outputs and decide which changes to accept. This

is especially valuable when a model update improves recall while introducing a new kind of false positive. Without run level provenance, the resulting graph changes are hard to explain.

Idempotency（幂等性）means that repeating an ingestion operation does not create unintended duplicates. A practical key might combine document version, source span, relation type, normalized arguments, and extraction policy. The key must reflect the semantics of the record: two separate experiments in one document should not merge merely because they share the same method and dataset. Idempotency therefore requires both technical identifiers and a clear definition of what counts as the same assertion.

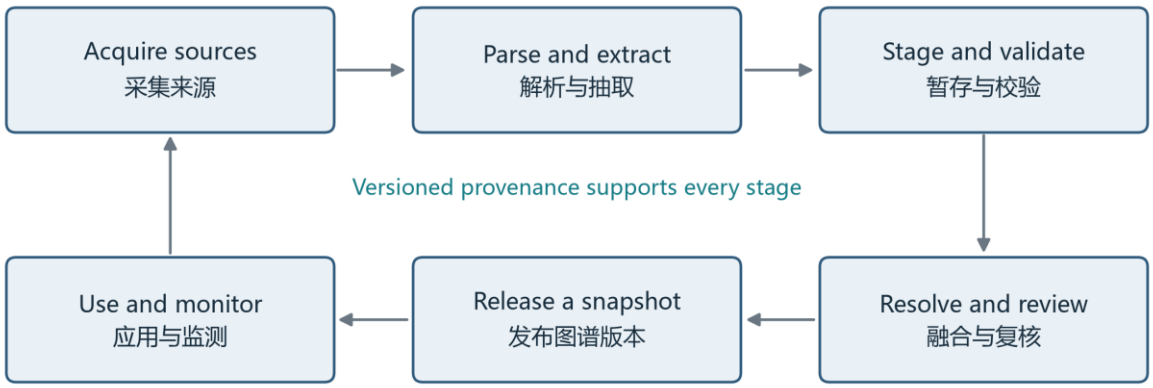


Figure 5. The knowledge graph lifecycle includes revision and monitoring. Provenance and versioning support every transition.

5 3 Fusion conflict and review

Knowledge fusion（知识融合）combines information from different sources while preserving its origin. A simple attribute such as a publication year can still conflict because a preprint, online publication, and issue date differ. A resolution policy should identify which date the application needs rather than choosing the most common value. For source dependent attributes, the graph may retain several values with qualifiers and expose a preferred value for a specific view.

Duplicate detection should distinguish repeated reporting from independent evidence. A journal article and its preprint may be versions of one study. A review article may summarize that study without adding a new experiment. A social media post may repeat the review. If all three are counted as separate supporting studies, the graph creates a false impression of consensus. Modeling publication lineage and evidence origin helps prevent this error, though ambiguous cases may still require review.

Human review should have a defined scope. A reviewer may confirm that an extraction matches the source without endorsing the source's scientific conclusion. Another reviewer may assess methodological quality. The system should record these as different activities. Otherwise, a generic “reviewed” label can be interpreted too broadly by downstream users. The same principle applies to automatic checks: passing a datatype validation is not equivalent to passing a scientific reliability review.

5 4 Updates deletions and repair

Incremental updates should identify which derived records depend on a changed source. If a document is corrected, its claims, summaries, embeddings, and cached answers may all need revision. A dependency graph（依赖图）can track these relationships. The graph itself then becomes part of the maintenance mechanism: it tells the system which outputs may have become stale. Dependencies should be recorded when artifacts are created, because reconstructing them later is often unreliable.

Deletion requires a policy for original content, extracted representations, backups, and derived artifacts. Removing a raw file while leaving its text in an embedding store does not fully remove the information from the system. Similarly, deleting a claim while retaining a community summary that repeats it can continue to mislead users. A repair process should propagate changes across every relevant representation and record the state of that propagation.

A failed update should not leave the accepted graph halfway between versions. Small related changes can use a transaction（事务）; larger imports may use a staged release and switch the active snapshot after validation. The choice depends on database capabilities and data volume. The important requirement is that users can identify which coherent version they are querying. An explicit snapshot identifier is often more informative than a vague “last updated” timestamp on the homepage.

5 5 Construction as a measurable service

Pipeline monitoring should include source freshness, extraction coverage, unresolved identities, validation failures, review backlog, and repair time. These measures describe whether the system is keeping up with its knowledge obligations. Node and edge counts alone are weak indicators. A rising edge count could reflect successful enrichment, duplicate imports, or a model that overproduces speculative relations. Operational metrics should connect to the competency questions established during schema design.

The proposed service publishes a small release report with each graph version. It lists changed sources, accepted and rejected assertions, schema changes, and known limitations. For a personal project, this report can be a simple file. For an organization, it may feed a dashboard and audit process. Its purpose is to make knowledge maintenance visible and repeatable, so that confidence in the graph rests on observable practices rather than on the apparent sophistication of the extraction model.

6 Knowledge storage and query architecture

6 1 Logical models and physical systems

Knowledge storage has several layers. The logical model describes entities, relationships, and semantics. The physical database determines how records are indexed, traversed, and updated. An application interface decides which operations users or agents can perform. These layers need not map one to one. A graph can combine a document repository, relational metadata, a graph database, and a vector index while presenting a coherent knowledge interface.

RDF stores organize triples or quads and commonly expose SPARQL. Labeled property graphs（标签属性图）represent nodes and relationships with labels or types and attached properties. The choice affects how qualifiers, identifiers, and query patterns are expressed. A property graph can make relationship attributes convenient for application development. RDF can make shared vocabularies and standardized interchange

convenient. Neither choice automatically ensures better performance, easier reasoning, or higher data quality. Those outcomes depend on workload, implementation, and governance.

The Research Evidence Graph uses an explicit Claim node even in a property graph because claims have their own provenance, review status, and lifecycle. This makes the model easier to translate between storage technologies. A simpler application might store an edge directly when the assertion is uncomplicated and rarely needs independent metadata. The design decision should follow the information's lifecycle rather than a preference for either maximum normalization or minimum node count.

Table 2. Logical models and their practical tradeoffs

Aspect	RDF	Labeled property graph
Core structure	Triples or quads with IRIs and literals	Nodes and typed relationships with properties
Query language	SPARQL	Implementation specific; Cypher is one example
Semantics	Can use standardized entailment regimes	Usually application defined; explicit rules can be added
Qualification	Named graphs or explicit assertion and event patterns	Relationship properties or explicit assertion and event nodes
Design test	Preserve identity, context, provenance, and required query meaning	The same semantic test applies

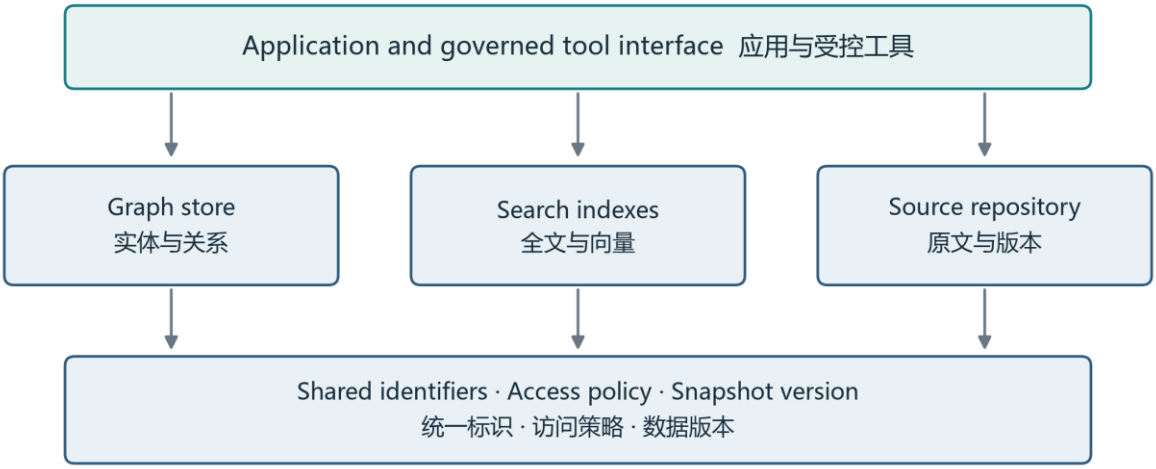


Figure 6. A hybrid architecture separates graph structure, retrieval indexes, and source documents while sharing identifiers and policy.

6 2 Query meaning and workload

SPARQL supports graph pattern matching, filtering, aggregation, optional patterns, and other operations over RDF data [9]. A SPARQL query can ask for papers connected to a method through evaluations and then filter by dataset. The query's answer depends on the dataset and any configured entailment regime. Standardization provides a shared language, but applications still need to define named graph usage, identity conventions, and whether inferred statements are included.

Cypher expresses property graph patterns through nodes and directed relationships. Its MATCH clause selects data matching a pattern, and subsequent clauses can filter or transform results [46]. In the proposed local implementation, an evidence retrieval query follows Method to Evaluation to Claim to SourceSpan. The path is deliberately bounded and typed. This makes the intended connection clear and limits expansion. An unrestricted traversal across every relationship can return a large but semantically incoherent neighborhood.

Query design should begin with selectivity. An exact method identifier or document key is usually a better starting point than scanning every node. Indexes accelerate suitable lookups, but their benefit depends on the query planner and data distribution. A common operational mistake is to create many indexes without measuring representative queries. The appropriate evidence is a query plan and observed latency on the actual workload, including realistic data sizes and concurrent users.

6 3 Hybrid storage and source authority

A document store preserves raw and parsed source material. A graph store preserves explicit relationships and identity. A full text index supports lexical matching. A vector index supports similarity retrieval over text or other representations. A relational or analytical store may hold large numeric tables and experiment metrics. Each system contributes a different access pattern. The application should identify which representation is authoritative for each kind of information and how derived copies are refreshed.

For example, the source repository is authoritative for the text of a paper version. The graph is authoritative for the accepted mapping between that version and its extracted claims. The vector index is a derived retrieval aid. If an embedding is stale, the system should regenerate it from the authoritative text; it should not treat the old vector as an independent record of truth. Stable identifiers and version tags are the links that make this separation manageable.

Hybrid retrieval can combine exact entity matching, lexical search, and vector similarity before expanding through graph relations. The order should depend on the question. A query containing a verified DOI can begin with exact identity. A vague conceptual question may begin with text similarity and then resolve candidate entities. The system can retain the retrieval route so that later evaluation shows whether an error came from initial retrieval, linking, graph expansion, or answer generation.

6 4 Transactions consistency and concurrency

A transaction groups related changes so that they succeed or fail together under the database's supported guarantees. Ingestion can create an Evaluation, connect its arguments, and attach its provenance in one transaction. If only part of that operation succeeds, the graph may contain an orphan record that looks valid to a simple query. Validation should therefore include structural integrity, while transactions protect the process that establishes that integrity.

Concurrent writers create additional challenges. Two extraction workers may attempt to create the same canonical entity. A uniqueness constraint on a stable key can prevent duplication, but the application must handle the conflict correctly and retry when appropriate. A generated title is not a reliable key. Nor should workers merge arbitrary records merely to avoid a conflict. The identity policy and the concurrency policy must agree about which records are allowed to collapse.

Distributed representations may update at different times. A graph transaction can complete before the search index has refreshed. The application should define whether temporary inconsistency is acceptable and how users can recognize it. A versioned retrieval request can require matching source and index versions for high assurance tasks. A casual exploration interface might accept a short delay. The important point is to

make the consistency requirement explicit instead of assuming that every component changes simultaneously.

6 5 Scale and partitioning

Graph workloads are sensitive to topology. A small number of high degree nodes can dominate traversal cost, while a large graph with selective bounded queries can remain manageable. Partitioning by department or geography may reduce storage and access scope, but relationships crossing partitions can become expensive. Replication can improve read availability while increasing the work required to maintain consistent updates. These tradeoffs should be tested with the actual relationship patterns rather than inferred from total node count alone.

Analytical computation often uses a projection（图投影）of the operational graph. A citation analysis might select only Paper nodes and CITES relationships. A collaboration analysis might derive an undirected graph of coauthors. Each projection changes the meaning of edges and may discard qualifiers. Its definition, source snapshot, and weighting rules should be stored with the result. Otherwise, two centrality scores with the same name can be mistakenly compared even though they were calculated on different graphs.

Large text, images, and binary files generally belong in an appropriate object or document store with graph references, rather than being duplicated in every connected node. This design keeps traversal focused on structure while preserving access to evidence. It also simplifies retention and versioning. The graph should store enough metadata to locate and authenticate the source, including a content hash when exact reproducibility matters.

6 6 Security backup and portability

Access control must apply to both records and derived views. A user who cannot access a confidential experiment should not receive a summary that reveals its result through an unrestricted graph neighborhood. Filtering only the final answer is too late if restricted information has already influenced retrieval or generation. The proposed architecture checks access during candidate retrieval, traversal, and evidence assembly. It also records which policy version governed the request.

Backup quality is demonstrated by restoration. A useful recovery exercise restores the graph, source references, schema, and indexes sufficiently to answer representative questions. Exporting triples or nodes alone may omit identity rules, procedures, or embeddings required by the application. Portability therefore includes both data and interpretation. An exchange package should document identifiers, relationship meanings, datatypes, provenance fields, and version assumptions.

Storage architecture is successful when it preserves meaning while meeting operational requirements. Fast traversal is useful, but an evidence graph also needs correct identity, coherent snapshots, recoverable updates, and understandable access rules. These properties make the stored knowledge dependable enough for the computations and agents discussed in the following chapters.

7 Knowledge computation and graph algorithms

7 1 What computation contributes

Knowledge computation（知识计算）is a broad term. In this report it includes querying explicit relationships, calculating structural properties, learning representations, and evaluating derived measures. These operations can reveal useful patterns without establishing new logical truths. A centrality score can identify a structurally prominent paper, but it does not prove that the paper is correct. A cluster can identify densely connected entities, but it does not prove that they belong to a naturally occurring category. The interpretation comes from the modeled relationships and the question being asked.

Every computation should begin with a graph specification. The specification states which node and edge types are included, whether direction matters, how weights are defined, how duplicate edges are handled, and which time interval is covered. In a research graph, treating CITES, CONTRADICTS, and AUTHORED_BY as one generic edge would mix different kinds of meaning. The resulting score might be mathematically well defined yet scientifically difficult to interpret. A reproducible analysis therefore records the projection as carefully as the algorithm.

7 2 Traversal reachability and shortest paths

Traversal（遍历）visits connected records according to a rule. Breadth first search finds shortest paths measured by the number of edges in an unweighted graph. Weighted shortest path methods solve a different problem: they minimize a sum of edge costs under the algorithm's assumptions. A cost might represent travel time, processing effort, or another quantity. It should not be chosen merely because the graph already contains a numerical attribute.

Suppose a dependency graph connects a research claim to the studies and datasets on which it relies. Reachability can answer which claims might be affected by correcting a dataset. The reachable set is a list of potentially affected records, not a judgment that every conclusion is invalid. A review process still needs to examine how each claim uses the dataset. This distinction allows a graph algorithm to support a decision without pretending to complete the whole decision.

The report's synthetic directed graph has four nodes, A, B, C, and D, and five edges: A to B, A to C, B to C, C to A, and D to C. From D, the shortest path to B is D to C to A to B, with length three. There is no directed path from A to D. An undirected rendering would hide that asymmetry and suggest a connection the directed model does not allow. Figure 7 preserves the arrows to make the distinction visible.

7 3 Adjacency matrices and walk counts

An adjacency matrix（邻接矩阵）records which node pairs are connected. For the synthetic graph, rows represent source nodes and columns represent target nodes in the order A, B, C, D. An entry of one denotes an edge. Multiplying the adjacency matrix by itself counts two step walks in this unweighted graph. The entry for D to A becomes one because D to C to A is available. The entry for A to A is also one because A to C to A returns to the start.

A walk（游走）can revisit nodes; a simple path cannot. This distinction matters when matrix powers are interpreted. Counting walks of length three does not count only three edge simple paths. It can include repeated vertices and cycles. Moreover, a two edge walk that combines different relationship types may

have no useful semantic interpretation. Matrix operations preserve the chosen algebraic structure, but they do not automatically supply the meaning of a composed relation.

The synthetic matrix calculation was checked with Wolfram using MatrixPower. Its purpose is pedagogical: it shows exactly what a graph computation returns and what it leaves unspecified. A practical system can use sparse matrices for large projections, but numerical efficiency does not change the interpretation. The report does not treat walk counts as probabilities, causal effects, or independent pieces of evidence.

7 4 A reproducible PageRank example

PageRank（页面排名算法）assigns a stationary distribution to a random walk with teleportation. The original work developed the idea for ranking web pages [28]. The same mathematical construction can be applied to other directed graphs, provided that the interpretation of a transition is clear. This report uses a damping factor of 0.85 and uniform teleportation across the four synthetic nodes. Each node divides its outgoing transition probability equally among its outgoing edges. There are no dangling nodes in this example.

$$\mathbf{r} = (1-\alpha)\mathbf{v} + \alpha P^{\mathsf{T}}\mathbf{r}, \quad \alpha = 0.85, \quad \mathbf{v} = \tfrac{1}{4}\mathbf{1}$$

In the displayed equation, r is the column vector of scores, P is the row stochastic transition matrix, and v is the uniform teleportation vector. Solving the linear system gives A = 0.3725268513, B = 0.1958239118, C = 0.3941492369, and D = 0.0375000000. The scores sum to one. Wolfram LinearSolve independently returned the same values and an exactly zero stationary residual in the rational calculation [47]. A local numerical calculation is included in the report's reproducibility checks.

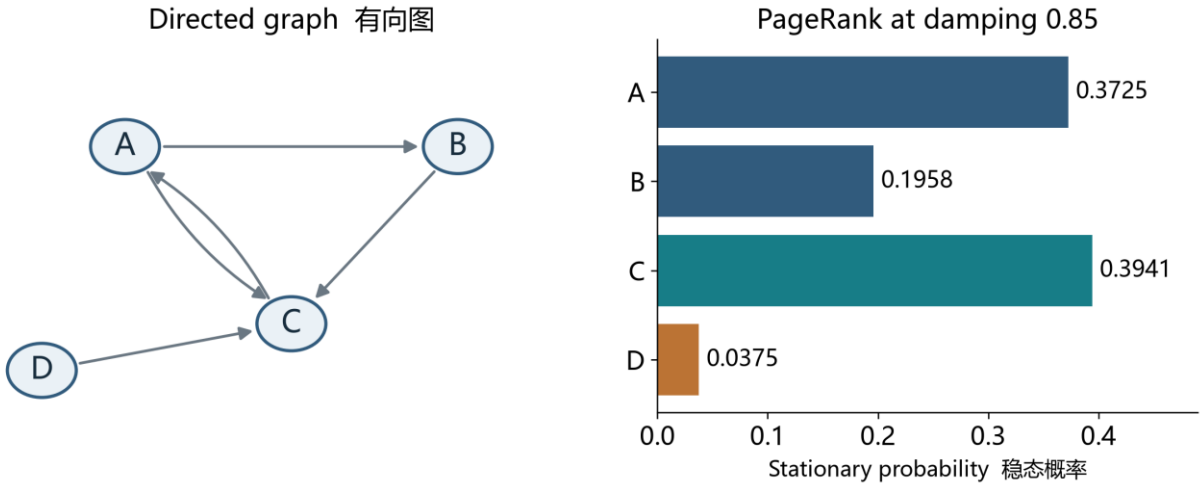


Figure 7. Synthetic directed graph and its PageRank distribution at damping 0.85. Scores were checked with Wolfram and a local linear system calculation; they are not empirical research results.

Node C receives the highest score because it receives transitions from A, B, and D, while A benefits from C's outgoing edge. D has no incoming links, so its score is its teleportation share. These are properties of this specific projection and transition rule. If nodes represented papers, the scores would describe structural prominence under the model. They would not measure experimental validity, methodological rigor, or the social value of the research.

Personalized PageRank changes the teleportation distribution to emphasize selected seed nodes. That can be useful for retrieving a neighborhood relevant to a user's question. The result remains sensitive to seed selection, relation filtering, and graph completeness. A node can receive a low score because it is poorly connected in the database, not because it is irrelevant in reality. A retrieval system should therefore combine structural ranking with lexical or semantic evidence when coverage is uneven.

7 5 Communities and similarity

Community detection（社区发现）groups nodes according to a structural objective. The Leiden algorithm addresses connectivity problems that can occur with Louvain and provides guarantees under its formulation [29]. For a research graph, communities may help organize a large citation network into navigable regions. The labels assigned to those regions still require interpretation. A community could reflect a method, a language group, a historical period, or citation practices rather than a coherent scientific subject.

Resolution settings and projection choices can change the grouping. A system should report these choices and test whether the interpretation survives reasonable alternatives. For exploratory browsing, approximate groupings can be useful. For a claim about the structure of a discipline, stronger validation is needed. Comparing clusters with expert labels, examining boundary cases, and checking stability over time can reveal whether the algorithm has found a useful analytical unit.

Similarity measures also depend on representation. In an undirected version of the synthetic graph, B has neighbors A and C, while D has neighbor C. Their Jaccard similarity is one divided by two, or 0.5. This number expresses overlap between two neighbor sets. It is not a probability that B and D should be connected. A learned link prediction model may use such overlap as a feature, but additional assumptions and validation are needed before converting it into a recommendation.

7 6 Representation learning and graph neural networks

Knowledge graph embeddings（知识图谱嵌入）map entities and relations to numerical representations so that a scoring function can distinguish plausible triples. TransE models relations as translations between entity vectors [21]. ComplEx uses complex valued representations to model a broader range of relational patterns [23]. RotatE represents relations through rotations in complex space [24]. These models differ in their inductive biases, so the relation patterns present in a dataset affect which assumptions are useful.

A small synthetic illustration makes the scoring idea concrete. Let an entity vector be (1, 0), a relation vector be (0, 1), and two candidate targets be (1, 1) and (3, 1). Under a Euclidean TransE distance, the first candidate has distance zero and the second distance two. The first therefore receives the more favorable score under this chosen geometry. These vectors were manually constructed for explanation; they were not learned from data and do not demonstrate predictive accuracy.

Graph neural networks（图神经网络）update representations by combining information from neighboring nodes and edges. R-GCN adapts graph convolution to relational data through relation dependent transformations [25]. NBFNet develops a learned path based framework inspired by a generalized Bellman–Ford formulation [26]. These approaches show how topology and relation semantics can enter learning. Their outputs still require evaluation under the deployment setting, particularly when new entities or relation types appear after training.

Table 3. Representative relational learning methods

Method	Main modeling idea	Interpretive limit
TransE [21]	Relation as vector translation	Plausibility score is not a factual proof
DistMult [22]	Tri-linear scoring	Symmetric score limits asymmetric modeling
ComplEx [23]	Complex valued factorization	Requires task specific training and evaluation
RotatE [24]	Relation as complex rotation	Learned relational patterns remain predictions
R-GCN [25]	Relation specific message passing	Quality depends on available neighborhoods
NBFNet [26]	Learned generalized path computation	Explanatory paths do not imply causal effects

7 7 Reproducibility and interpretive limits

An analytical result should include graph snapshot, projection, algorithm version, parameters, random seed when applicable, and output interpretation. For approximate algorithms, the team should record tolerances and convergence criteria. For learned models, it should record training data, negative sampling, validation policy, and checkpoint. These details enable another analyst to determine whether a changed result reflects new knowledge, different modeling choices, or numerical variation.

The strongest use of graph computation is often to narrow a human or automated investigation. It can prioritize nodes for inspection, expose disconnected records, identify dependencies, or rank candidate evidence. Its role becomes clearer when the system names the question it solves. A graph score should enter an application with an explicit interpretation and a tested decision rule, rather than acquiring authority merely because it was computed over a knowledge graph.

8 Knowledge reasoning and the status of conclusions

8 1 Several meanings of reasoning

Knowledge reasoning（知识推理）includes several distinct activities. Deduction（演绎推理）derives consequences from premises and rules. Induction（归纳推理）learns patterns from observations. Abduction（溯因推理）searches for explanations that could account for observations. Probabilistic inference estimates beliefs under a specified model. Causal inference（因果推断）concerns effects of interventions under causal assumptions. In graph based machine learning, “reasoning” may refer to predicting a missing edge or answering a multi hop question. The term should therefore be qualified whenever the type of guarantee matters.

The Research Evidence Graph uses separate statuses for asserted, validated, derived, and predicted statements. A derived statement has a rule and premises that explain why it follows. A predicted statement has a model and score that explain how it was ranked. An asserted statement has a source. A validated statement has passed a specified acceptance process. One statement can have several of these relationships, but the system should never silently replace one status with another.

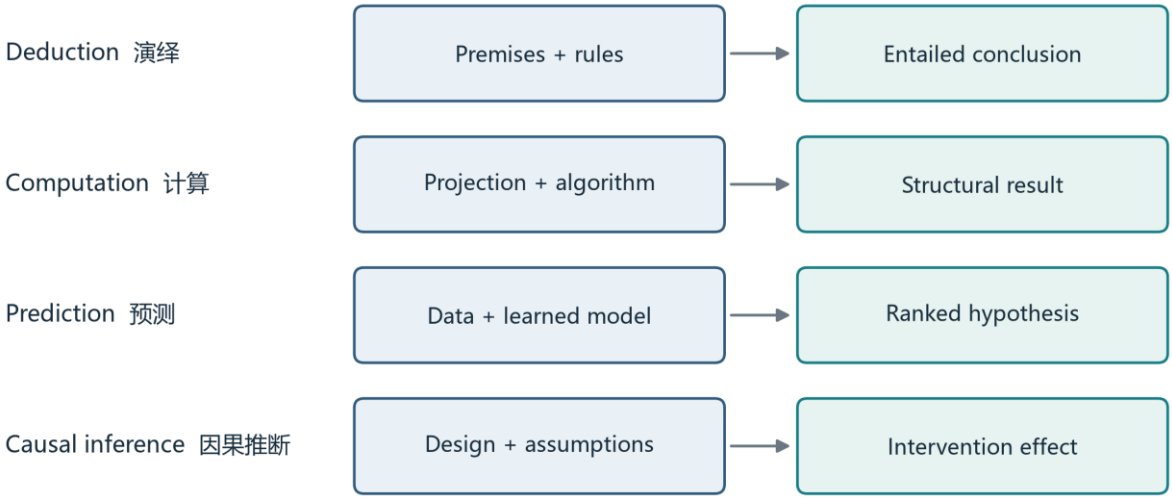


Figure 8. Four operations with different inputs and interpretations. A structural result or ranked hypothesis does not have the status of a logical or causal conclusion.

8 2 Deduction with explicit rules

Suppose the schema states that every Dataset is a ResearchResource and Dataset D is a Dataset. A reasoner can derive that D is a ResearchResource. This is a simple class hierarchy inference. A different rule might state that if a document reports an evaluation and the evaluation uses a dataset, the document has an evidence dependency on that dataset. The conclusion follows only because the application has explicitly defined this dependency rule. Ordinary graph connectivity alone does not establish it.

Rule based reasoning can be implemented by forward chaining（前向链推理）, which derives consequences from available facts, or backward chaining（反向链推理）, which works from a query toward supporting premises. Materialization（物化推理）stores derived results in advance; query rewriting or on demand reasoning computes relevant consequences when needed. The preferred method depends on update frequency, query patterns, and the rule language. A frequently queried stable hierarchy may suit materialization, while rapidly changing facts may make selective computation more attractive.

Every derived result should retain a justification（推导依据）or a reproducible route to one. If Dataset D is reclassified, the system must know which derived statements depend on that classification. Simply storing the derived triple without its provenance makes correction difficult. A truth maintenance process can retract conclusions whose support no longer holds, while retaining conclusions that have another valid derivation. This is a practical requirement for an evolving graph, even when the underlying rules are simple.

8 3 Open world semantics and query boundaries

Under open world reasoning, failing to prove a statement does not prove its negation. If the graph contains no record of a paper using Dataset D, the correct answer may be “not known in this graph.” A user asking “Which papers do not use D?” needs a stronger completeness assumption. The application might restrict the query to papers whose methods have been fully reviewed. It should communicate that scope so the negative result remains interpretable.

SPARQL entailment regimes specify how query evaluation can incorporate different semantics beyond simple graph matching [52]. A deployment should document whether it uses plain stored triples, RDFS consequences, an OWL profile, or custom rules. Two endpoints containing similar explicit data can return different answers because their reasoning settings differ. A reproducible query therefore needs both the data snapshot and the entailment configuration.

Closed world checks are appropriate for some operational contracts. If every accepted Evaluation is required to contain a metric, a missing metric is a validation failure within that application. This does not imply that the real world experiment had no metric; it means the stored record is incomplete under the contract. Keeping validation and logical entailment separate lets a system use both without confusing their purposes.

8 4 Rule mining and uncertain regularities

Rule mining（规则挖掘）searches a graph for recurring relational patterns. AMIE 3 improves the efficiency of mining rules from knowledge bases [27]. A mined rule can identify a useful regularity, but its measured confidence is not a logical guarantee. For example, two authors who repeatedly publish together may often share an institution, yet coauthorship does not logically imply common employment. Promoting that regularity to a strict rule would create false affiliations.

A responsible workflow separates discovery, validation, and deployment. A mined pattern first becomes a candidate rule. Domain experts assess whether it expresses a meaningful regularity or a data collection artifact. Held out data tests its predictive value. The application then decides whether to use it for suggestions, anomaly detection, or a probabilistic model. Only a relation that is justified as a logical commitment should become a hard inference rule.

Observed confidence is sensitive to missing data. If a graph records some predicates more completely than others, a rule can appear weaker or stronger because of the recording process. The system should ask what is known about completeness before interpreting a confidence score. A high score among well documented entities may not transfer to poorly documented entities. This concern connects rule mining to the broader issue of selection bias in knowledge acquisition.

8 5 Link prediction as hypothesis generation

Link prediction（链接预测）ranks missing relations as candidates for further investigation. A model may learn that certain methods are often evaluated on certain dataset families. It can then suggest a likely missing evaluation link. That suggestion is useful for finding omissions, but it does not demonstrate that an experiment actually occurred. If the target relation describes an event, external evidence is needed before the predicted edge enters the accepted factual layer.

DistMult illustrates how a scoring function can impose a structural limitation. Its standard tri-linear score is symmetric in the head and tail entity vectors for a fixed relation [22]. This can be useful for symmetric patterns but cannot by itself distinguish every asymmetric relation. The observation is mathematical, not a claim that the model is universally ineffective. A deployed system may contain additional components or constraints, and evaluation must reflect the full implementation.

$$s(h, r, t) = \sum_{k=1}^{d} h_k r_k t_k = s(t, r, h)$$

Negative sampling（负采样）creates candidate triples used as training negatives, often by replacing an argument in an observed triple. In an incomplete graph, some sampled negatives may actually be true but unrecorded. The training procedure therefore operates with assumptions about missingness. A research report should describe those assumptions rather than treating all absent edges as verified falsehoods. The same issue affects evaluation when a model predicts a plausible relation missing from the benchmark annotations.

Transductive evaluation（传导式评估）typically tests new links among entities seen during training.

Inductive evaluation（归纳式评估）tests generalization to previously unseen entities or graphs under a defined setting. A model that succeeds in one setting may fail in the other. For an organizational graph that receives new projects every week, performance on unseen entities is a central requirement. Choosing a benchmark merely because it is popular can miss that requirement entirely.

8 6 Probability calibration and explanation

A ranking score is not automatically a probability. A model can rank correct candidates highly while producing numerical scores that poorly reflect actual correctness frequencies. Calibration（校准）studies the agreement between predicted confidence and observed outcomes; Guo and colleagues demonstrated calibration problems in modern neural networks and evaluated corrective methods [30]. In a graph application, calibration must be checked on the relevant relation types and deployment distribution rather than assumed from a general model.

An explanation should identify the actual basis of an answer. A rule based conclusion can expose premises and a rule. A retrieval answer can expose source passages. A learned prediction can expose influential paths or features, but such an explanation may be an approximation to model behavior. A path displayed after prediction is not necessarily the path that caused the prediction. The interface should label the explanation type and avoid presenting an attractive graph path as a formal proof.

8 7 Causal claims require causal assumptions

A causal graph represents assumptions about how variables influence one another. A generic knowledge graph can contain causal claims, but it does not become a valid causal model merely by labeling an edge CAUSES. Causal inference requires a defined intervention question, assumptions about confounding and structure, and an identification or estimation strategy [50]. A centrality measure or an association pattern cannot substitute for those requirements.

In the research example, papers reporting that a method improves an outcome provide evidence about a claim. Whether the claim is causally supported depends on study design, comparison conditions, and alternative explanations. A graph can organize those features and retrieve relevant evidence. It cannot repair

an uncontrolled comparison simply by connecting it to other studies. This limitation should appear near any causal interpretation, especially when an agent summarizes mixed experimental and observational evidence.

8 8 A practical reasoning architecture

The proposed architecture routes questions to the appropriate operation. Exact factual questions use accepted assertions and sources. Taxonomic questions use a configured deductive layer. Exploratory discovery uses rankings and predicted links clearly labeled as hypotheses. Causal questions retrieve study designs and assumptions for a separate analysis. This routing can be implemented through templates and typed tools before introducing an elaborate autonomous planner.

The answer record contains the operation type, input snapshot, evidence, uncertainty, and any rules or models used. If the system cannot satisfy the required evidence standard, it returns an incomplete answer or asks for a narrower question. This behavior preserves the usefulness of the graph without assigning every output the same authority. The central design principle is that the status of a conclusion should remain visible from its production through its presentation and any subsequent action.

9 Applications and the conditions for value

9 1 Search discovery and public knowledge

Knowledge graphs can support search by connecting a query to identified entities, their aliases, categories, and relationships. A search for an institution can retrieve its current and historical names; a search for a research topic can connect related terminology; a query about a paper can expose authors, datasets, and follow up work. The benefit is strongest when the relationships help resolve ambiguity or move between information types. Simply adding graph storage behind a conventional keyword search does not establish that benefit.

Wikidata is a major example of a collaboratively maintained structured knowledge base. Its design supports statements about identifiable items and a multilingual setting [31]. It demonstrates that public knowledge infrastructure requires community processes as well as data structures. For an application using external knowledge, the relevant lesson is to retain identifiers and statement context while accounting for change. Importing an external label without recording its source and retrieval date makes later reconciliation harder.

An enterprise discovery system can use similar ideas at a smaller scale. A question about “the retention policy” may refer to customer data, employee records, or research material. Connecting policies to owners, scopes, versions, and affected systems can make the ambiguity explicit. The interface can then ask a focused clarification or present scoped alternatives. This is often more useful than returning several semantically similar documents without explaining why they differ.

9 2 Scientific and biomedical research

Scientific knowledge graphs connect publications, datasets, methods, measurements, and hypotheses. Their value is not limited to finding papers. They can help identify which results use the same data, which claims depend on a particular method, and where contradictory findings occur under different conditions. A careful graph can also reveal missing metadata that prevents meaningful comparison. Such gaps are research findings about the evidence base, even when the system cannot produce a confident answer to the original scientific question.

Hetionet integrates biomedical knowledge to prioritize candidates for drug repurposing [32]. PrimeKG assembles a multimodal graph intended to support precision medicine analyses [33]. These projects illustrate how diverse biological and clinical concepts can be connected for computational investigation. Their existence does not imply that every predicted relationship is clinically validated. In this report, they serve as research infrastructure examples, not as medical recommendations or evidence that a graph based prediction alone should determine treatment.

The distinction between a hypothesis and an intervention is especially consequential in scientific applications. A candidate link can guide literature review or laboratory planning. Moving from that candidate to a claim about an effective intervention requires appropriate empirical evidence. The graph can make the evidence chain easier to inspect and can help avoid repeating known experiments. Its role is to support the scientific process, not to replace the standards by which scientific claims are evaluated.

9 3 Organizational knowledge and decision support

Organizations often maintain separate records for people, projects, documents, systems, and procedures. A graph can connect these records around questions such as who owns a process, what depends on a system, and which projects use a dataset. The key difficulty is usually identity and lifecycle alignment. A project code may change, a team may reorganize, and a document may remain online after its policy is superseded. Without temporal and ownership modeling, integration can make stale information easier to find rather than more reliable.

A useful deployment starts with a bounded workflow. For example, an internal research office might need to identify all projects affected by a dataset license change. The graph connects dataset versions to projects, publications, and responsible people. A query produces a review list with source evidence. Success can be measured by completeness of the affected project list and time required to confirm it. This is more concrete than a general promise to “connect all organizational knowledge.”

Graph based decision support should distinguish descriptive information from operational authority. A graph may record who appears to own a system, but the action service should verify current ownership before allowing a consequential change. The record may be stale or contested. This distinction becomes even more important when an agent moves from answering questions to initiating tasks. Knowledge access and action permission are related, but they are not the same control.

9 4 Recommendation and personalization

A knowledge graph can enrich recommendation with relationships among users, items, topics, creators, and contexts. In a learning application, a resource can connect to concepts, prerequisites, difficulty, and learning objectives. A recommendation can then explain that the resource addresses a concept the learner has not yet mastered or provides a missing prerequisite. The usefulness of the explanation depends on the quality of those relationships and on evidence about the learner, rather than on the mere presence of a graph path.

Personalization introduces risks of narrow feedback loops. If users only see resources connected to their previous behavior, the system may reinforce a limited view of their interests. A graph can help introduce diversity by traversing different relation types, but diversity should be a measured objective. A recommendation system should evaluate relevance, novelty, coverage, and user outcomes. It should also allow users to correct inferred preferences and distinguish explicit goals from behavioral guesses.

Educational prerequisite graphs require particular caution. A curriculum author may assert that concept A should precede concept B, while actual learners may follow several viable routes. Representing prerequisites

as universal physical laws would be inappropriate. The graph can instead attach the curriculum, learner population, and evidence to the relationship. An adaptive agent can then offer a route with a stated rationale and revise it in response to observed learning, rather than treating the graph as an inflexible hierarchy.

9 5 Supply chains engineering and risk analysis

Engineering applications often have naturally relational questions: which components depend on a supplier, which requirements are verified by a test, and which changes affect a product configuration. A graph can represent these dependencies with version and configuration context. The same component may behave differently in two assemblies, so a dependency should not be assumed to apply universally. Configuration specific relationships help prevent a broad traversal from producing an exaggerated impact list.

Risk analysis can use connected components, paths, and shared attributes to prioritize cases for investigation. Shared addresses or devices may be informative, but they can also have innocent explanations. The graph should preserve the underlying observations and the uncertainty of the interpretation. A connected cluster should not be presented as proof of misconduct. The appropriate application output may be a ranked review queue with explicit reasons and an opportunity for correction.

For critical engineering workflows, the graph can support traceability from requirement to design, implementation, and verification evidence. This is a stronger basis for value than a generic network visualization. A reviewer can inspect whether every requirement has supporting evidence and whether a changed component invalidates an earlier test. The model must still account for versions, applicability conditions, and authority. A test for an obsolete configuration cannot automatically certify the current one.

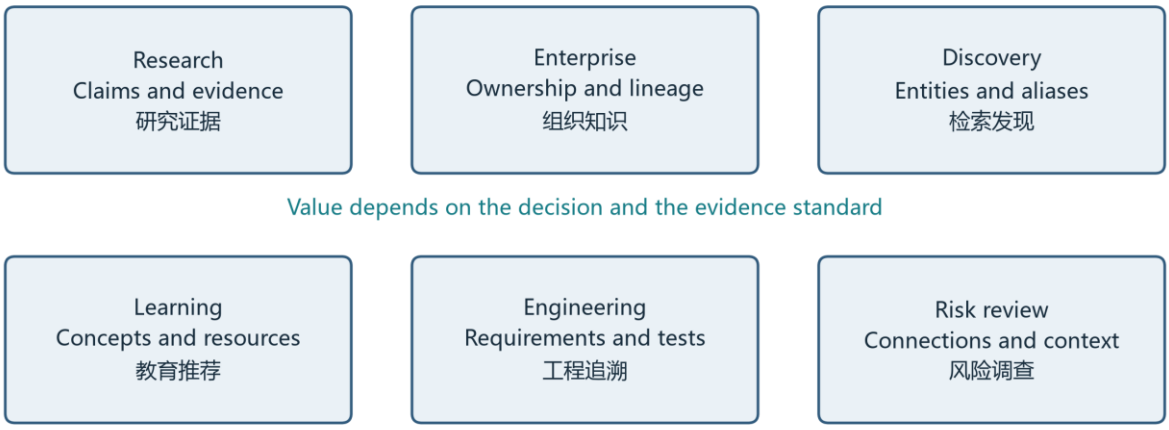


Figure 9. Application areas organized by the relationships that make graph modeling useful. These are task categories, not measured market shares.

9 6 Evaluate value against the task

Application value should be measured against a specific alternative. If a graph helps analysts find affected projects faster than a spreadsheet and document search workflow, the comparison should use the same cases and count verification time. If a graph based recommender improves learning, the evaluation should examine relevant learning outcomes rather than only clicks. If a graph enriches retrieval, answer quality should be assessed independently of the system's own explanations.

Costs include initial modeling, source access, extraction, review, storage, updates, and user training. Benefits may include faster discovery, fewer duplicate records, better traceability, or improved decisions. A credible business case measures the relevant costs and outcomes locally. This report does not assign universal return on investment figures because they depend on workload and organizational conditions. The graph earns its role when its explicit relationships solve a recurring problem better than a simpler approach at an acceptable total cost.

10 Language models retrieval and GraphRAG

10 1 Complementary representations

Large language models（大语言模型）represent regularities learned from training data in their parameters. Knowledge graphs represent selected information explicitly through identifiers, relationships, and context. These forms of representation support different operations. A language model can interpret flexible questions and generate readable text. A graph can support precise identity, structured constraints, and traceable updates. Combining them is useful when an application needs both linguistic flexibility and explicit control over evidence.

Pan and colleagues describe several broad directions for unifying language models and knowledge graphs, including graph enhanced language models, language model assisted graph tasks, and closer joint integration [35]. The report adopts this broad distinction while focusing on an operational question: what information crosses the interface, and what guarantees survive the crossing? Turning a precise graph result into prose can lose qualifications. Turning prose into graph records can introduce unsupported precision. Both directions need validation.

10 2 Retrieval augmented generation

Retrieval augmented generation（检索增强生成）conditions an answer on retrieved external information. Lewis and colleagues developed an influential approach combining retrieval with sequence generation for knowledge intensive tasks [34]. In an application, retrieval can update the evidence available to a model without changing all of its parameters. Yet retrieval alone does not guarantee factuality. The system may retrieve irrelevant material, omit a decisive source, or generate an answer that goes beyond the retrieved evidence.

A conventional text retrieval pipeline can be a strong baseline. It searches document chunks, reranks candidates, and provides selected passages to a language model. If the question asks for a definition clearly stated in one document, a graph may add little value. Graph based retrieval becomes more attractive when the answer depends on connecting entities across documents, following typed relationships, or aggregating evidence at a broader level. The workload should determine whether that additional structure is worth constructing and maintaining.

10 3 Several designs called GraphRAG

GraphRAG（图谱检索增强生成）is a family of designs rather than one universally specified algorithm. A system may retrieve explicit triples from an established graph, build a graph from documents, use communities for summarization, or combine graph neural representations with textual evidence. These

variants differ in construction cost, query behavior, and error modes. A research comparison should identify which architecture is being evaluated instead of treating all graph augmented systems as equivalent.

Edge and colleagues propose a graph based approach to query focused summarization, using an extracted graph and community summaries to support questions about a corpus as a whole [36]. The contribution addresses a different retrieval need from finding one local fact. It suggests why precomputed summaries can help global questions, while also making construction and summary maintenance part of the system's cost. The report does not infer from this work that every factual question benefits from global community summarization.

LightRAG combines graph structure with retrieval mechanisms designed to access information at different levels [41]. G-Retriever investigates retrieval augmented generation for textual graphs [39]. LEGO-GraphRAG analyzes graph based retrieval through modular components and empirical comparisons [42]. These studies collectively motivate careful decomposition of the pipeline. Each component can improve one dimension while increasing latency, token use, or dependence on the quality of the constructed graph.

Table 4. Selected research designs and the question each addresses

Work	Architectural emphasis	Status
GraphRAG [36]	Corpus level summaries from extracted graph communities	2024 preprint
Think-on-Graph [37]	Interactive graph path exploration	ICLR 2024
RoG [38]	Planning, retrieval, and graph grounded reasoning	ICLR 2024
G-Retriever [39]	Retrieval for textual graph questions	NeurIPS 2024
LightRAG [41]	Graph supported retrieval at different levels	Findings of EMNLP 2025
LEGO-GraphRAG [42]	Modular analysis of graph retrieval design	PVLDB 2025
Youtu-GraphRAG [43]	Agentic integration of construction and retrieval	2025 preprint

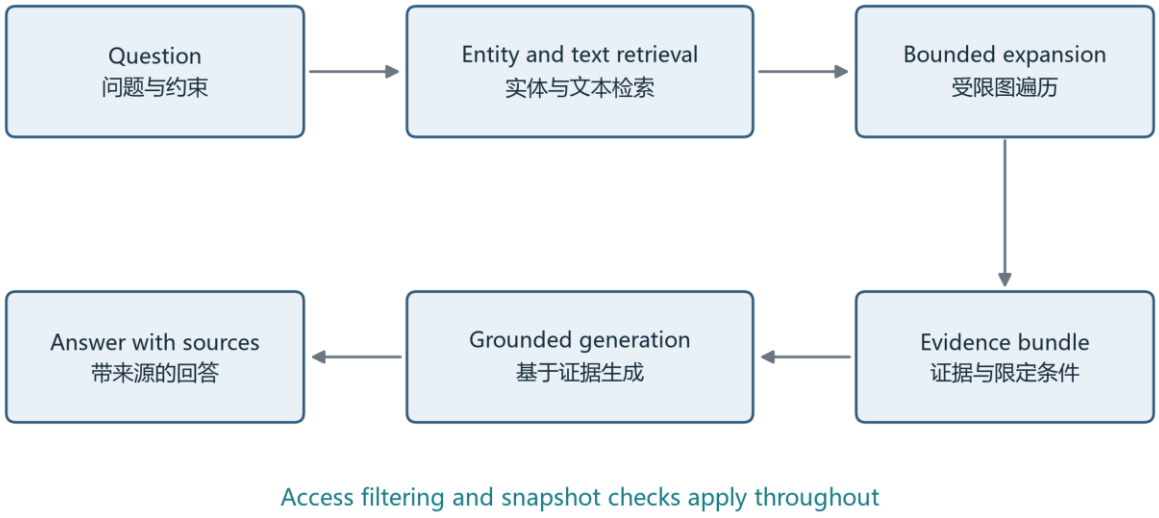


Figure 10. A proposed hybrid GraphRAG workflow. Source evidence and qualifiers are passed to the generator after bounded graph expansion.

10 4 A proposed hybrid retrieval pipeline

The proposed Research Evidence Graph assistant first interprets the question and identifies constraints such as method, dataset, date, and evidence standard. It then generates candidate entities through exact identifiers, aliases, lexical search, and semantic retrieval. Candidate resolution is explicit: if “GraphLens” maps to several entities, the system does not silently choose one. It either uses available context to disambiguate or returns a scoped clarification.

After selecting seeds, the retriever follows an allowlisted set of relationship types with bounded depth and result size. For a question about experimental support, it follows evaluations and source claims rather than arbitrary coauthorship links. It collects both the structured records and the source passages that support them. The evidence bundle contains identifiers, source versions, relevant qualifiers, and access decisions. This bundle is the grounding material for answer generation.

A reranker can prioritize evidence by relevance and completeness, but the answer policy should preserve conflicting results. If the top ranked study supports a claim while another valid study contradicts it, a summary that omits the contradiction may be misleading. The system can reserve part of its context budget for disagreement and uncertainty. That is an application design choice to be tested, not an assumption that a relevance model will automatically discover the right scientific balance.

10 5 Global questions and summary maintenance

A question such as “What are the main methodological disagreements in this collection?” requires synthesis across many documents. Community summaries can provide an intermediate representation that is cheaper to inspect than all raw text for every query. Their usefulness depends on whether the communities and summaries preserve the relevant distinctions. A cluster formed by citation proximity may not align with the disagreement the user cares about. The system may need several organizing views or a query specific aggregation step.

Summaries should carry provenance and dependency links to their inputs. When a source changes, the system should mark affected summaries as stale. It should also avoid counting a summary and its underlying sources as independent evidence. A summary is a derived artifact, not another study. This is a recurring source of accidental evidence inflation when multiple retrieval layers are combined without explicit lineage.

The context window remains a resource constraint even when the graph is large. Retrieving every adjacent node can overwhelm the generator with loosely relevant information. A bounded evidence selection process should optimize for answer requirements, diversity of support, and contradiction coverage. More retrieved tokens do not necessarily mean more knowledge. They can also obscure the decisive evidence and increase the chance that the model combines incompatible contexts.

10 6 Faithfulness and fair comparison

Faithfulness（忠实性）asks whether an answer is supported by the supplied evidence. Correctness asks whether the answer is accurate under the task's reference standard. These measures can diverge: an answer can be accidentally correct while unsupported, or faithfully summarize a source that is wrong. A research assistant should be evaluated on both. Citation presence alone is insufficient; the cited passage must support the specific statement attached to it.

A fair comparison holds the language model, source collection, date cutoff, and answer budget constant where possible. It compares text retrieval, hybrid text retrieval, graph retrieval, and any community summary

variant on different question classes. Local fact questions, multi source questions, temporal questions, and global synthesis questions should be analyzed separately. Otherwise, an overall average can conceal a substantial improvement on one class and a regression on another.

Costs should include graph construction, updates, and query time. A system that spends heavily before the first query may be economical for a stable corpus with many repeated questions, but expensive for a rapidly changing collection with few queries. Reporting only query latency can hide that difference. The appropriate design is the one that delivers the required evidence quality under the actual lifecycle cost, not the one with the most elaborate retrieval diagram.

11 Agents and knowledge graphs

11 1 What makes a system an agent

An AI agent is a system that selects and executes actions toward a task using observations, internal state, and available tools. Autonomy varies: some systems follow a fixed workflow with a model at one step, while others choose among tools and revise plans over several steps. A knowledge graph can provide structured state, memory, constraints, and evidence for either design. The presence of a graph does not by itself make a system autonomous, and autonomy does not imply that every action should be delegated to a model.

ReAct explores interleaving reasoning and acting in language model based systems [40]. For knowledge graph applications, the relevant architectural idea is that a system can inspect information, choose another action, and revise its approach based on the result. The user does not need access to a model's private internal reasoning for this process to be auditable. A useful audit trail records observable tool calls, returned data, selected evidence, and actions taken.

The Research Evidence Graph agent has a bounded task: answer research questions with traceable evidence and prepare proposed graph updates for review when new documents are provided. It is not authorized by default to publish conclusions, alter source files, or make external commitments. Defining that scope allows the tool interface and evaluation to focus on concrete behavior. It also makes failure recovery easier because the system knows which actions were legitimate parts of the task.

11 2 Graphs as agent memory

Agent memory（智能体记忆）can be divided by function. Episodic memory（情景记忆）records interactions and events. Semantic memory（语义记忆）stores relatively stable concepts and facts. Procedural memory（程序性记忆）stores reusable methods or workflows. A graph can link these forms without making them interchangeable. A record that the user once asked about a topic is an interaction event; it is not proof that the user has a permanent preference or that a scientific claim about the topic is true.

Memory should include provenance, time, scope, and correction mechanisms. A preference inferred from behavior should be distinguishable from an explicit user instruction. A temporary task constraint should not become a permanent global rule. In a multi user environment, memory also needs an owner and access policy. Graph links can make cross task retrieval powerful, but they can also spread an incorrect or unauthorized inference farther than a simple isolated note would.

The proposed memory model stores user approved preferences separately from extracted domain claims and execution logs. Retrieval selects the relevant memory type for the current task. If an agent needs a writing preference, it does not search scientific evidence as though the two were equivalent. If a user corrects a preference, the system updates the active record and preserves enough history to explain the change. This design keeps personalization interpretable.

11 3 Graph guided planning and exploration

Think-on-Graph treats a language model as an agent that explores a knowledge graph and searches for promising reasoning paths [37]. Reasoning on Graphs separates planning, retrieval, and reasoning around graph grounded relation paths [38]. These methods demonstrate ways to use explicit graph structure during question answering. Their reported benchmark results apply to their experimental settings; they do not establish unrestricted reliability for arbitrary organizational graphs or action taking agents.

A practical planner can use a schema to propose a sequence of typed operations. For the question “Which accepted claims depend on the corrected dataset version?”, it can resolve the dataset, traverse dependency links, filter claim status, and retrieve source evidence. The system can verify each operation against a tool contract. If the requested relation does not exist, the planner should revise the plan or explain the gap rather than generating a query against an imagined schema.

Graph exploration should be bounded by depth, number of results, time, and cost. Without bounds, a high degree node can cause a large expansion that adds little relevant information. A stopping rule should depend on evidence sufficiency and task requirements. The agent can stop when it has the required sources, when additional retrieval produces no useful evidence, or when it reaches a budget. Stopping because an answer sounds fluent is not a reliable evidence criterion.

11 4 Typed tools and controlled execution

A typed tool（类型化工具）accepts structured arguments and returns a defined result. Instead of exposing unrestricted database commands to every agent step, the proposed interface includes operations such as resolve_entity, get_claim_evidence, find_dependencies, and propose_assertion. These names describe application capabilities, not a requirement for a particular framework. A tool implementation can use Cypher, SPARQL, or another backend while enforcing the same contract.

For exploratory analysts, a controlled query tool may still be appropriate. It should distinguish read operations from writes, validate the query structure, bind parameters, enforce timeouts, and restrict accessible procedures. Read only should be enforced by the execution environment or database role where possible, rather than by asking the model to promise not to write. The tool response should also identify truncation so the agent does not mistake a result limit for complete coverage.

Write operations need stronger contracts because they change future knowledge. A proposed assertion should contain canonical arguments, relation type, evidence, qualifiers, and expected graph version. Validation can then check whether the write is admissible and whether the underlying state changed since the proposal was prepared. An idempotency key prevents a retried request from duplicating the record. These controls support reliable automation without requiring a human to supervise every harmless read.

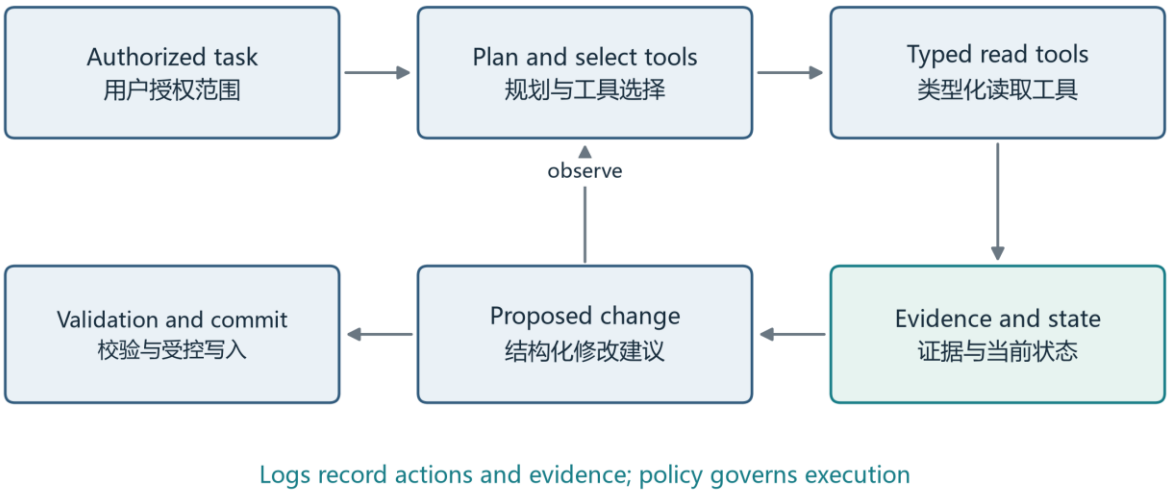


Figure 11. A proposed agent architecture separating authorized tasks, read tools, evidence, and validated writes. The observe loop supports plan revision.

11 5 Evidence is not an instruction channel

Documents and graph properties can contain text that looks like an instruction. A retrieved passage might say “ignore previous rules” or request that the system send data elsewhere. In a research workflow, that text is source content to be analyzed, not authority over the agent. The execution layer should preserve this boundary. A graph does not eliminate prompt injection（提示注入）risk if untrusted text is later inserted into a model context without distinguishing data from authorized instructions.

The proposed agent labels tool outputs by origin and uses only the user's authorized task and system policy to choose actions. It can quote or summarize instructions found in a document when they are relevant to the research question, but it does not execute them merely because they appear in retrieved evidence. This is especially important for write tools and external services. Provenance helps identify where content came from; permission checks determine what the agent may do with it.

11 6 Planning knowledge and action authority

A graph can describe possible actions, prerequisites, and expected effects. For example, a workflow graph may state that a dataset must be validated before an analysis is released. The agent can use that information to plan. The action service must still verify the actual current prerequisite before executing the release. A stored statement that validation occurred may be outdated, and the agent's retrieved snapshot may lag behind the operational system.

This separation addresses a common design weakness: treating a semantic relationship such as CAN_APPROVE as sufficient authorization. The graph may help discover the relevant role, but the authoritative permission system should decide whether the current actor can perform the action. Similarly, a path showing that a user belongs to a team is not enough if membership expired or if the permission applies only to a different project. Time, scope, and policy version must be evaluated at execution.

11 7 Recovery learning and multiple agents

An agent should record the state needed to recover from a failed operation. If retrieval times out, it can retry a bounded read. If a write response is lost, it should check the idempotency key before repeating the write. If the schema changes during a task, it should reload the relevant contract. Recovery behavior is easier to test when actions have explicit preconditions and results rather than being embedded in long free form scripts.

Several agents may share a knowledge graph, but shared storage does not guarantee coordination. They may use different snapshots, attach conflicting claims, or duplicate work. A shared task record can identify ownership, status, and accepted outputs. Optimistic concurrency checks can detect stale updates, while provenance identifies which agent produced a candidate. A human or deterministic policy may still be needed to resolve semantic conflicts. Agreement among several models is not independent empirical confirmation when they share sources and assumptions.

Learning from interaction should be selective. Successful completion of one task does not establish that every intermediate inference was correct. The system can retain verified corrections, reusable tool outcomes, and user approved preferences while leaving speculative reasoning outside persistent factual memory. This prevents a temporary mistake from becoming a trusted premise in later tasks. Memory quality should be evaluated over sequences of tasks, where accumulation and correction effects become visible.

11 8 Recent directions and deployment criteria

Think-on-Graph 2.0 investigates graph guided retrieval with richer information access [44]. Youtu-GraphRAG proposes a more integrated agentic approach linking graph construction and retrieval [43]. These are useful examples of research directions, but this report cites their identifiable preprint records and does not treat their performance claims as independently reproduced results. Their main relevance here is architectural: construction, schema, retrieval, and reasoning increasingly interact rather than operating as isolated modules.

An agent should be deployed only at the level of autonomy supported by task specific evaluation. A research assistant that reliably retrieves evidence may still be poor at updating a canonical graph. A system that writes valid records may still be unfit to publish conclusions automatically. Evaluation should distinguish read quality, planning quality, update quality, permission compliance, and recovery. The knowledge graph can make these dimensions observable, provided that the design preserves the status and origin of every consequential piece of information.

12 Evaluation quality and governance

12 1 Evaluate the full chain

Knowledge graph evaluation（知识图谱评估）must connect component quality to application outcomes. A system can have accurate entity extraction but poor identity resolution, fast queries but incomplete evidence, or fluent answers that omit important contradictions. Evaluating each component separately identifies defects, while evaluating the complete workflow shows whether those defects matter to users. Neither level replaces the other. The proposed evaluation plan uses both a diagnostic suite and a set of representative research tasks.

The diagnostic suite covers parsing, entity detection, linking, relation extraction, qualifier preservation, validation, retrieval, and answer generation. Each stage receives inputs representative of those it will see in operation. Testing a relation extractor only with perfect entity labels can measure its isolated capability, but it does not measure deployed performance after entity errors. Both results can be useful if the report clearly identifies which setting produced each score.

12 2 Extraction metrics with a worked example

Precision（精确率）is the fraction of predicted positive items that are correct. Recall（召回率）is the fraction of reference positive items recovered. F1 is their harmonic mean. Suppose a synthetic evaluation contains 90 correct extracted relations, 10 incorrect extracted relations, and 30 missed reference relations. Precision is 90 divided by 100, or 0.90. Recall is 90 divided by 120, or 0.75. F1 is approximately 0.8182. These are illustrative counts, not results from a real extractor.

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2PR}{P + R}$$

The definition of a correct relation must be fixed before scoring. An experiment may require exact entity identities and a relation label, while another also requires the correct dataset version and polarity. These evaluation contracts produce different numbers. A score should therefore be accompanied by the matching rule and the annotation policy. Otherwise, two systems may appear comparable while one is being rewarded for much less complete extraction.

Micro averaging（微平均）aggregates counts across categories, allowing frequent classes to dominate.

Macro averaging（宏平均）averages class level scores, giving rare classes more influence. Both can be informative. A research graph with many easy authorship relations and few difficult contradiction relations can achieve a high micro score while performing poorly on its most important analytical relation. Reporting per relation results helps prevent that failure from disappearing into an overall average.

12 3 Ranking and retrieval metrics

Mean reciprocal rank（平均倒数排名）averages the reciprocal of the rank assigned to a correct answer. For three synthetic queries with correct answer ranks one, two, and four, the mean reciprocal rank is approximately 0.5833. Hits at one is one third; Hits at three is two thirds. These metrics describe ordering under the defined candidate set. They do not measure whether the answer is sufficiently supported by evidence or whether the candidate set includes every valid answer.

Knowledge graph completion benchmarks sometimes use filtered ranking to avoid penalizing a model for ranking another known correct answer ahead of the test target. A report should state the filtering policy and which known triples were used. Filtering can make benchmark comparison more consistent, but it does not solve unknown truth in an incomplete graph. Real deployment also requires a decision about when to show a candidate, when to abstain, and whether the relation is suitable for automatic acceptance.

Retrieval evaluation should include evidence recall at the chosen context budget, not merely whether a relevant document appears somewhere in a long list. A system may find the correct paper but omit the table footnote needed to interpret its result. For multi source questions, evaluation should check whether the

retriever finds all essential evidence components and relevant contradictions. This gives a more realistic view of what the generator can reasonably answer.

12 4 Leakage and distribution shift

Data leakage（数据泄漏）occurs when evaluation information improperly influences training or system design. In graph tasks, inverse relations and near duplicate records can create subtle leakage. If a test triple has an equivalent inverse stored in the training graph, predicting it may require little genuine generalization. If versions of the same paper appear in both training and test sets, extraction performance can benefit from repeated wording. Split design should account for these dependencies.

Temporal evaluation is useful when the application will predict or retrieve information that arrives later. The model and graph should use only information available before the cutoff, and evaluation should reflect the later period. This includes auxiliary text and embeddings: a graph split is not truly temporal if the language model input contains future descriptions. The report should identify unavoidable exposure risks and avoid claiming a perfect historical simulation when the training data of a foundation model are unknown.

Distribution shift（分布漂移）can arise from new domains, languages, document layouts, or source practices. A system trained on research abstracts may perform poorly on tables and technical reports. Evaluation should reserve realistic out of domain cases and monitor changes after deployment. A falling confidence score may reveal some shifts, but confident mistakes remain possible. Periodic review of randomly sampled accepted outputs is therefore a useful complement to automated monitoring.

12 5 Faithfulness and human evaluation

Human evaluation should use clear criteria and evidence access. Reviewers can separately assess answer correctness, source support, completeness, handling of uncertainty, and usefulness. A single “good answer” score mixes these dimensions and makes disagreement difficult to interpret. Reviewers should not be shown system identities when that information could bias judgment. When several reviewers evaluate the same sample, disagreements should be analyzed rather than averaged away without explanation.

Language model judges can help scale preliminary evaluation, but they should be calibrated against human judgments for the task. They may favor fluent language, share biases with the generating model, or overlook unsupported details. A practical workflow uses automatic checks for clear structural failures, model based assessment for triage, and human review for a representative sample and consequential cases. The evaluation report should state which conclusions depend on each method.

Uncertainty intervals should reflect dependence in the data. Many claims from one document are not independent observations. Resampling individual claims can give overly narrow intervals if errors cluster by document. A document level or source family level resampling strategy may better reflect the unit of variation. The chosen statistical method should match the study design, and small sample results should be presented as preliminary rather than made precise through excessive decimal places.

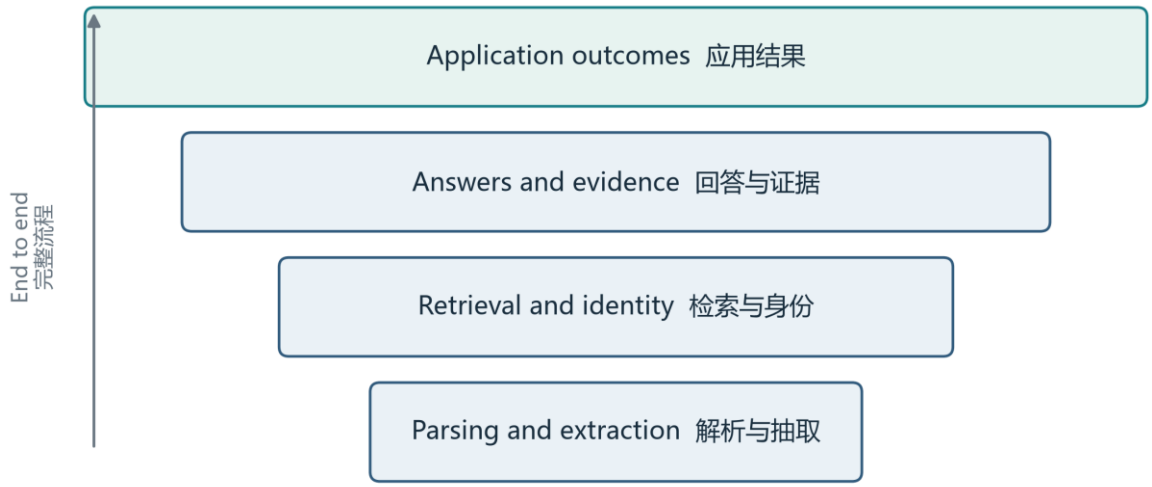


Figure 12. Evaluation combines component diagnostics with complete application outcomes. Higher level success depends on lower level quality but is measured separately.

12 6 Governance and evidence quality

Governance defines responsibility for the graph's meanings, records, and uses. Each important dataset should have an owner, permitted uses, update policy, and correction route. FAIR principles emphasize findability, accessibility, interoperability, and reusability in scientific data stewardship [49]. Applying these principles to a graph requires more than publishing an endpoint. Users need understandable identifiers, metadata, access conditions, and enough context to reuse the information correctly.

Quality dimensions can conflict. A very strict acceptance policy may improve precision while reducing coverage. A broad public graph may increase reuse while creating privacy or licensing problems for particular sources. The appropriate policy depends on the task and the rights associated with the data. A governance record should explain the tradeoff and identify who can revise it. This gives users a basis for interpreting omissions instead of assuming that the graph represents all available knowledge.

The proposed release decision combines minimum quality requirements with a realistic maintenance plan. It asks whether the system answers its competency questions accurately, whether important errors can be corrected, and whether the team can keep sources current. A graph that passes a one time demonstration but has no update owner is not ready to become a dependable source for an agent. Sustainable operation is part of quality, not an optional activity after deployment.

Table 5. Evaluation measures should match the failure being tested

Layer	Measure or check	What it does not establish
Extraction	Precision, recall, qualifier accuracy	Scientific truth of the source
Identity	Linking accuracy, merge and split errors	Completeness of the entity catalog
Retrieval	Evidence recall at a fixed budget	Faithfulness of generated prose
Answer	Correctness and citation support	Permission to execute an action
Agent	Task completion, policy compliance, recovery	Reliability outside the tested scope

Layer	Measure or check	What it does not establish
Operation	Freshness, repair time, lifecycle cost	Value without a useful task

13 A complete Research Evidence Graph implementation

13 1 Scope and fictional source collection

This chapter develops an original implementation design for a small research team. The aim is to answer evidence questions about retrieval methods while preserving provenance and allowing corrections. The initial collection contains three fictional documents and a small set of explicitly invented records. It is intentionally small enough to inspect by hand. The design can later scale, but the first objective is to make every transformation understandable and testable.

Document P1, dated February 2025, reports an evaluation of GraphLens and PathSearch on Dataset D version 1. Document P2, dated May 2025, repeats the comparison on Dataset D version 2 with a different outcome. Document P3, dated June 2025, corrects a metadata field in P1 and explains that one reported score used a different query subset. None of these documents represents an actual publication. The fictional names prevent the worked example from being mistaken for an empirical claim about a real method.

The competency questions are specific. Which results compare the two methods on the same dataset version and metric? Which claim depends on the corrected query subset? Which sources support a statement of improvement, and which use conditions that prevent direct comparison? The system should answer these questions with source references and should decline to produce an overall winner unless an aggregation policy and comparable evidence justify one.

13 2 Minimum schema and records

The minimum schema contains Document, SourceSpan, Person, Method, Dataset, Evaluation, and Claim. A Document has a local identifier, title, version, date, and content hash. A SourceSpan has a locator and a reference to its document version. An Evaluation connects the compared methods, dataset version, metric, and reported measurements. A Claim expresses the interpreted result and links to the relevant Evaluation and source evidence. Review status belongs to the claim record, while bibliographic identity belongs to the document.

The synthetic measurements use mean reciprocal rank on a stated query set. P1 reports 0.42 for GraphLens and 0.38 for PathSearch on D version 1. P2 reports 0.40 for GraphLens and 0.41 for PathSearch on D version 2. These invented values support two context specific comparisons. They do not justify averaging the four numbers without considering query sets, uncertainty, and evaluation protocols. The correction in P3 makes the comparability question even more important.

Table 6. Invented evaluation records for the worked case

Document	Dataset	GraphLens	PathSearch	Context
P1	D version 1	0.42	0.38	Query subset later qualified by P3
P2	D version 2	0.40	0.41	Different evaluation context
P3	Correction	—	—	Revises interpretation of P1

For P1, the graph can store an attributed claim that GraphLens had the higher reported score under the specified evaluation. It should not store the universal claim that GraphLens is better than PathSearch. For P2, it stores the corresponding result under the second dataset version. A comparison view displays both contexts and the correction notice. The graph's value is that it retains the structure needed to interpret the disagreement instead of forcing the documents into one unqualified edge.

13 3 Ingestion and validation

The ingestion process saves immutable source versions and extracts their document structure. It identifies candidate entities and links them to the local method and dataset registry. The extractor emits candidate evaluation records with source locators and explicit missing fields. It does not fill absent confidence intervals or query counts from general knowledge. A validator checks identifiers, numeric datatypes, relation directions, and the presence of required context.

The first human review focuses on the meaning of each comparison and the correction's scope. The reviewer confirms whether P3 changes the numerical value, the population to which it applies, or only a descriptive field. That distinction determines which records need revision. If the correction affects comparability but not the original measurement, the graph retains the original result and adds a qualification. Deleting the entire result would lose useful history and overstate the correction's effect.

The accepted release receives a snapshot identifier and a small manifest listing source versions, schema version, extraction configuration, and review decisions. The search index and any generated summaries record that snapshot. A later reprocessing run can propose changes without altering the currently accepted view. This makes model experimentation possible while preserving a stable basis for users' ongoing research.

13 4 Querying the accepted evidence

A bounded query can retrieve accepted claims associated with a selected dataset and their source passages. The following Cypher example illustrates the intended pattern for a property graph implementation. It assumes the schema just described and a bound dataset identifier parameter. It is a teaching example, not a command executed against the user's existing database. Database credentials, deployment paths, and private records are outside this report.

```
MATCH (c:Claim {status: 'accepted'})-[:BASED_ON]->(e:Evaluation)
      -[:USES_DATASET]->(d:Dataset {id: $dataset_id})
MATCH (c)-[:EVIDENCED_BY]->(s:SourceSpan)-[:IN_DOCUMENT]->(p:Document)
RETURN c.id, c.text, p.id, s.locator
ORDER BY c.id
```

The query returns evidence records rather than a free form conclusion. A rendering layer can display the claim, document, and locator together. If several source spans support one claim, the interface should group them without treating each span as a separate study. If access restrictions remove one source, the system must avoid implying that the remaining visible set is complete. The answer can state the scope of the accessible evidence and any resulting limitation.

A corresponding RDF implementation would express the same domain entities through IRIs and predicates and use SPARQL graph patterns. The semantic design should remain recognizable across both versions. The purpose of the example is to show that storage choice can change query syntax without changing the core evidence commitments. Translation should be tested for qualifiers, identity, and provenance rather than judged only by whether node counts match.

13 5 Agent interaction and correction

Suppose a user asks, “Does GraphLens outperform PathSearch?” The agent resolves the method identities and retrieves the two evaluations and correction. It should answer that the fictional evidence reports different outcomes under different conditions, then identify those conditions. It should explain that the available records do not establish a universal winner. A useful response can still be concise: it presents the two results, notes the correction, and identifies what additional comparison would be needed.

If the user asks for a graph update after providing P3, the agent prepares a proposed change containing the affected claim, evidence locator, qualification, and expected snapshot. The validator checks the proposal, and the configured review policy decides whether it can be accepted automatically. The write records the new state and its provenance. Any dependent summary is marked stale and regenerated before it is served as current evidence.

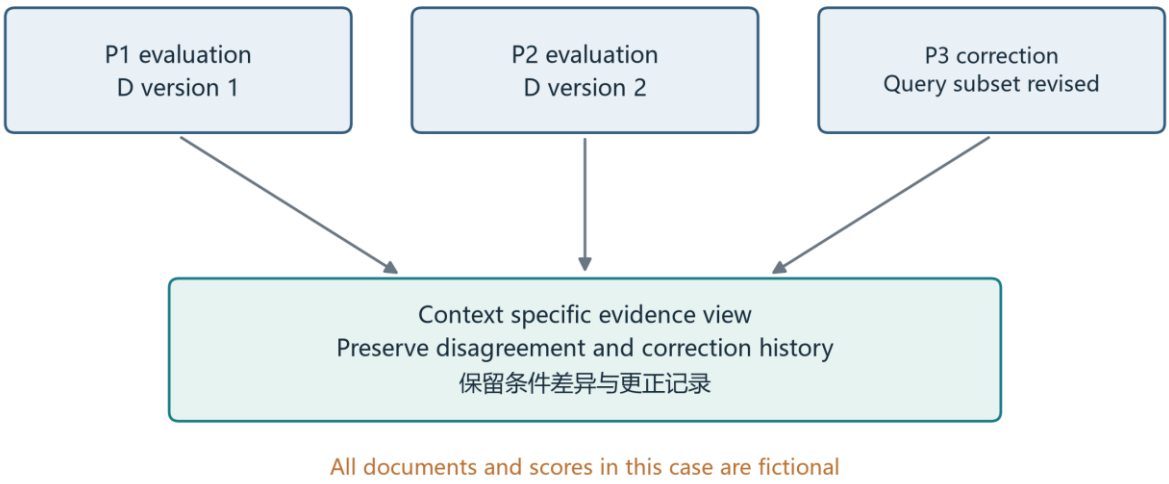


Figure 13. The fictional Research Evidence Graph preserves two evaluation contexts and a correction. No real method performance is asserted.

If an extraction run incorrectly links GraphLens to an unrelated product, the repair operates on the mention to entity mapping. The source remains intact. The system identifies claims and summaries that depended on the incorrect mapping, revises those records, and records the repair. This example demonstrates why reversible identity decisions are essential. A graph that had overwritten the source names and discarded mappings would make the same correction much harder.

13 6 Testing the implementation

The first acceptance test checks identity: P1's and P2's GraphLens mentions map to the intended method, while an unrelated same name entity remains separate. The second checks context: the dataset versions and query subsets remain distinct. The third checks evidence: every accepted comparison resolves to an exact source location. The fourth checks correction propagation: P3 updates the relevant view and invalidates its dependent summary without erasing unrelated results.

A retrieval test asks both a local factual question and an ambiguous global question. The local question should return the specified score and context. The global question should preserve disagreement and avoid an unsupported ranking. An access test removes permission to P2 and verifies that neither its direct result

nor a derived summary leaks into the answer. A recovery test retries an interrupted update and confirms that the idempotency key prevents duplication.

Performance testing comes after semantic correctness. The team measures ingestion time, review effort, query latency, and evidence assembly size on representative documents. It can then decide whether to add indexing, caching, or more advanced retrieval. Starting with these measurements keeps optimization tied to observed bottlenecks. A small reliable system provides a better foundation for expansion than a large demonstration whose correctness cannot be inspected.

13 7 A phased delivery plan

The first phase establishes scope, identifiers, a minimal schema, and a manually checked seed collection. The second automates parsing and extraction into staging, with review and error analysis. The third adds bounded retrieval and evidence based answers. The fourth adds controlled updates, correction propagation, and monitoring. More autonomous planning is introduced only after the earlier capabilities are measured. These are dependency based phases rather than fixed calendar promises.

Each phase has an exit criterion. The seed graph must answer the competency questions. Extraction must meet a locally chosen quality threshold on held out documents. Retrieval must recover the evidence needed for representative questions. Updates must preserve provenance and recover from failure. This plan provides an implementation path for a local Neo4j project while remaining applicable to RDF or another backend. The central deliverable is a trustworthy evidence workflow, with the database serving that workflow.

14 Research priorities and conclusions

14 1 Better benchmarks for changing knowledge

Knowledge graphs evolve, yet many benchmarks evaluate a static snapshot. Future evaluation should test correction, deletion, temporal scope, and newly arriving entities. A system should be rewarded for revising an answer when its evidence changes and for preserving the distinction between what was believed earlier and what is accepted now. Such benchmarks would examine a capability central to practical knowledge management but easy to miss in static link prediction scores.

Another priority is evidence dependence. Several sources may repeat one original claim, while apparently contradictory results may concern different conditions. Benchmarks that require identifying these patterns would better test research assistants than questions with one isolated answer triple. They would also encourage systems to model publication lineage, study context, and uncertainty. Evaluation should include cases where the appropriate output is a qualified answer or a request for additional evidence.

14 2 Multimodal and multilingual graphs

Multimodal knowledge graphs（多模态知识图谱）connect information from text, images, tables, audio, and other sources. The difficulty is preserving what each modality actually supports. An image may establish that an object is present but not its identity; a table may establish a measurement but require a caption to define the units. Aligning modalities should retain their source specific uncertainty rather than collapsing every observation into one confident entity record.

Multilingual graphs face a related problem of alignment. Equivalent words do not always imply equivalent concepts, and culturally specific categories may not map neatly into one universal hierarchy. Future systems

should support language specific labels, contextual mappings, and documented disagreements. For Chinese and English research work, terminology management is a practical starting point: retain both original expressions and the agreed concept identifier, while allowing the mapping to be revised.

14 3 Learning with constraints and explanations

Neural and symbolic methods can complement each other when their responsibilities are clear. Learned models can propose candidates and rank evidence; symbolic constraints can detect impossible types or enforce a declared rule. The research challenge is to combine them without mistaking a constraint violation for a factual error or a high model score for proof. Evaluation should measure both predictive utility and the correctness of the claimed explanations.

Explanations should become testable artifacts. A retrieved path can be checked against the graph; a rule derivation can be checked against its premises; a source citation can be checked against the passage. The stronger claim that an explanation accurately describes a model's internal mechanism requires additional evidence. Systems should expose the explanation they can verify rather than promising a deeper form of interpretability because the output includes a graph visualization.

14 4 Agents that maintain knowledge responsibly

Agents may increasingly help maintain graphs by identifying missing metadata, reconciling duplicates, suggesting schema extensions, and locating outdated claims. These tasks are attractive because their outputs can often be reviewed as concrete changes. A productive research direction is to measure how much correct maintenance work an agent completes per unit of review effort. That measure connects autonomy to a real operational benefit while keeping error correction visible.

Agent memory also needs evaluation over time. An assistant should retain useful verified information, forget or retire obsolete records, and avoid amplifying its own mistakes through repeated retrieval. Tests spanning many sessions can reveal failures that single question benchmarks miss. Shared graphs may make memory more inspectable, but they also create a durable route through which an incorrect assumption can influence many later tasks. Versioned provenance and correction are therefore central research topics.



Figure 14. A dependency based implementation path. Greater autonomy follows measured evidence quality, maintenance capability, and task evaluation.

14 5 Final synthesis

A knowledge graph is most valuable when explicit identity and relationships make important questions answerable with clearer evidence. Its effectiveness depends on the entire lifecycle: schema commitments, faithful extraction, identity resolution, storage, computation, reasoning, and maintenance. Each stage contributes a different kind of guarantee. A reliable system preserves those guarantees and their limits as information moves toward an application or an agent.

The practical route is to begin with a bounded question set, model the evidence required to answer it, and build a small inspectable graph. Automated extraction and learned prediction can expand coverage, while validation and provenance keep the expansion correctable. Graph retrieval and agents can then provide flexible access to the knowledge, provided that they preserve uncertainty and operate through controlled tools. This approach makes knowledge graphs a durable component of research and organizational work rather than a one time visualization or an unexamined layer of generated assertions.

Appendix A Reproducibility and interpretation notes

A 1 Exact computational specification

The numerical graph example uses an ordered node set A, B, C, D. Its directed edges are A to B, A to C, B to C, C to A, and D to C. No edges are inferred or added. The adjacency matrix has rows (0, 1, 1, 0), (0, 0, 1, 0), (1, 0, 0, 0), and (0, 0, 1, 0). The transition matrix divides each row by its outdegree, so the first row becomes (0, 0.5, 0.5, 0), while the other rows remain unchanged.

With damping 0.85 and four equally likely teleportation destinations, each node receives a teleportation contribution of 0.0375. The linear system is solved without rounding the input fractions. The final decimal values in Chapter 7 are rounded only for presentation. Wolfram's LinearSolve verified the stationary equation, and its PageRank documentation provides the interpretation of damping and personalization [47,48]. The report's build process checks the same solution numerically using the explicitly defined matrix, avoiding ambiguity about a software library's default graph orientation.

The square of the adjacency matrix has rows (1, 0, 1, 0), (1, 0, 0, 0), (0, 1, 1, 0), and (1, 0, 0, 0). These rows count two step walks. They are not a normalized probability matrix. A transition matrix squared would instead describe two step transition probabilities for the specified random walk without teleportation. Distinguishing the two matrices prevents a common mistake in which a walk count is presented as a probability merely because both calculations use matrix multiplication.

The undirected Jaccard example intentionally uses a different projection from PageRank. Arrow directions are removed, yielding the unique edges AB, AC, BC, and CD. Neighbor overlap is then computed on that undirected projection. The report explicitly changes the projection because comparing an undirected similarity score with a directed centrality score without explaining the change would obscure their meanings. Neither result is interpreted as a statistical estimate of a real world relationship.

A 2 Evidence selection and source status

The review selected sources to cover formal representation, extraction, storage, graph algorithms, reasoning, and language model integration. Foundational standards and primary research papers anchor specific technical claims. Broad surveys were used to orient the taxonomy and locate representative methods. Recent architecture papers were included when they illustrated a materially different approach, while numerical performance comparisons were omitted where this report did not reproduce or closely audit the experimental setup.

Consensus search results were followed by full record retrieval before citation. SciSpace identified additional graph retrieval and agent papers, and technical descriptions were checked against accessible original paper records. Wolfram supplied computational verification for the synthetic example. These tools supported discovery and checking; they are not substitutes for the cited publications. The bibliography distinguishes standards, published papers, preprints, and software documentation.

The date on a search index can refer to a preprint, an online publication, or a journal issue. Where versions differ, the reference identifies a usable publication record and may also link to an accessible preprint. A reader comparing results should consult the same version and experimental conditions. This report does not claim that its source list exhausts research available by the search date, and it does not rank methods as current state of the art.

A 3 A small design experiment for readers

A useful exercise is to represent the same fictional evaluation in two ways. First, create a direct edge from GraphLens to Dataset D labeled EVALUATED_ON. Second, create an Evaluation node linking the method, dataset version, metric, baseline, and source. Ask each model to answer which method had the higher score under a specified protocol. The direct edge is compact, but it lacks the context needed for the comparison unless that context is added elsewhere. The exercise reveals a modeling tradeoff through a concrete question.

Repeat the exercise after introducing a correction and a second evaluation. The better representation is the one that preserves both results and allows the affected context to change without rewriting unrelated history. RDF's introductory examples provide a useful starting point for expressing resources and statements, but the domain specific evaluation structure remains a design responsibility [51]. A property graph implementation should be tested against the same questions, so the comparison concerns preserved meaning rather than syntax preference.

A 4 Interpreting aggregate outcomes

Suppose a team evaluates two retrieval systems on 100 questions. System X answers 80 correctly and System Y answers 84 correctly. Those totals alone do not establish that Y is reliably better. The team needs to know which questions differ, whether the sample represents deployment, and whether both systems used the same evidence and budget. A paired analysis can use the per question outcomes because the systems were tested on the same items. Repeated questions derived from one document may also require grouping by source.

The same discipline applies to graph construction. A larger graph may contain more useful evidence, more duplicates, or more unsupported assertions. Counting nodes and edges cannot distinguish those outcomes. A meaningful experiment connects structural changes to task performance and review effort. For the Research Evidence Graph, a useful outcome is the proportion of answers with complete and correct supporting evidence, measured alongside the time required to maintain that quality. This turns graph development into a testable research and engineering process.

Appendix B English and Chinese terminology

The glossary gives the meaning used in this report. Translations are intended to support reading rather than impose a universal terminology standard. Some fields use alternative Chinese terms, especially for entity resolution, grounding, and reasoning. When a term has several meanings, the report states the relevant setting in the surrounding text.

Table 7. Terminology used in the report

English term	中文术语	Meaning in this report
Knowledge graph	知识图谱	An explicit representation of entities, relationships, and interpretable context.
Entity	实体	A referent that the system identifies persistently.
Mention	提及	A source expression that may refer to an entity.
Canonical entity	规范实体	The current shared representation to which source mentions are linked.
Relation	关系	A named connection with a defined meaning and direction.
Triple	三元组	A subject, predicate, and object statement.
Schema	模式	The categories, relationships, and constraints available to the model.
Ontology	本体	An explicit conceptual model that may include formal logical commitments.
Taxonomy	分类体系	A hierarchy organizing categories.
Competency question	能力问题	A question the modeled knowledge must be able to answer.
Provenance	溯源	Information about sources, activities, and responsibility behind a record.
Valid time	有效时间	The interval when a statement applies in the modeled world.
Transaction time	记录时间	When the information entered or changed in the system.
Named entity recognition	命名实体识别	Detection and classification of named text spans.
Entity linking	实体链接	Mapping a mention to a known entity or leaving it unresolved.
Entity resolution	实体消歧与归一	Reconciling identities across records and sources.
Coreference resolution	共指消解	Identifying expressions that share a referent in context.
Relation extraction	关系抽取	Recovering source expressed relations and essential qualifiers.
Knowledge fusion	知识融合	Combining source records while handling identity and conflict.
Labeled property graph	标签属性图	Nodes and typed relationships with attached properties.
RDF	资源描述框架	A standardized graph data model based on triples.
SPARQL	RDF 查询语言	A language for querying RDF data and related operations.
OWL	Web 本体语言	A family of ontology constructs with formal semantics.
SHACL	形状约束语言	A language for specifying validation constraints on RDF graphs.
Open world assumption	开放世界假设	Missing statements need not be false.

English term	中文术语	Meaning in this report
Closed world assumption	封闭世界假设	Absence is treated as false within a specified complete scope.
Entailment	逻辑蕴涵	A conclusion follows under the declared semantics and premises.
Deduction	演绎推理	Deriving consequences from premises and rules.
Induction	归纳推理	Learning general patterns from observations.
Abduction	溯因推理	Seeking plausible explanations for observations.
Causal inference	因果推断	Reasoning about interventions using explicit causal assumptions.
Graph projection	图投影	A selected or transformed graph used for a particular computation.
PageRank	页面排名算法	A stationary distribution under a random walk with teleportation.
Community detection	社区发现	Grouping nodes according to a structural objective.
Knowledge graph embedding	知识图谱嵌入	Numerical entity and relation representations used for scoring or learning.
Graph neural network	图神经网络	A learned model that combines information through graph structure.
Link prediction	链接预测	Ranking candidate missing relationships under a model.
Calibration	校准	Agreement between predicted confidence and observed correctness.
Retrieval augmented generation	检索增强生成	Generating an answer conditioned on retrieved external information.
GraphRAG	图谱检索增强生成	A family of retrieval and generation designs using graph structure.
Grounding	证据支撑	Connecting an output to the information used to support it.
Faithfulness	忠实性	Whether an output accurately reflects its supplied evidence.
Agent	智能体	A system selecting and executing actions toward a task.
Typed tool	类型化工具	An operation with structured inputs, outputs, and an enforced contract.
Idempotency	幂等性	Repeating an operation avoids unintended additional effects.
Data leakage	数据泄漏	Evaluation information improperly influences training or design.
Distribution shift	分布漂移	Deployment data differ from the conditions used to build or test a model.
Abstention	暂不判断	Withholding a conclusion when the required evidence is insufficient.
Snapshot	数据快照	A coherent identified version of the data used by a query or analysis.
Truth maintenance	真值维护	Tracking derivations so conclusions can be revised when support changes.

References

The following references identify the works and standards cited in the report. Web resources were checked during preparation on 1 October 2026. Preprints are labeled explicitly. Links identify the cited record or an accessible original version; they do not imply that this report reproduced the paper's experiments.

[1] Hogan, A., et al. (2021). Knowledge Graphs. ACM Computing Surveys, 54(4). Peer reviewed survey. Source record

[2] Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2022). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. IEEE Transactions on Neural Networks and Learning Systems, 33(2), 494–514. Peer reviewed survey. Source record

[3] Zhong, L., Wu, J., Li, Q., Peng, H., and Wu, X. (2024; online 2023). A Comprehensive Survey on Automatic Knowledge Graph Construction. ACM Computing Surveys, 56. Peer reviewed survey. Source record

[4] W3C (2014). RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. Source record

[5] W3C (2014). RDF 1.1 Semantics. W3C Recommendation. Source record

[6] W3C (2014). RDF Schema 1.1. W3C Recommendation. Source record

[7] W3C (2012). OWL 2 Web Ontology Language Primer, Second Edition. W3C Recommendation. Source record

[8] W3C (2017). Shapes Constraint Language, SHACL. W3C Recommendation. Source record

[9] W3C (2013). SPARQL 1.1 Query Language. W3C Recommendation. Source record

[10] W3C (2013). PROV-O: The PROV Ontology. W3C Recommendation. Source record

[11] W3C (2009). SKOS Simple Knowledge Organization System Reference. W3C Recommendation. Source record

[12] W3C (2012). OWL 2 Web Ontology Language Profiles, Second Edition. W3C Recommendation. Source record

[13] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural Architectures for Named Entity Recognition. NAACL HLT, 260–270. Peer reviewed paper. Source record

[14] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL HLT. Peer reviewed paper. Source record

[15] Wu, L., et al. (2020). Scalable Zero-shot Entity Linking with Dense Entity Retrieval. EMNLP. Peer reviewed paper. Source record

[16] Eberts, M., and Ulges, A. (2020). Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. ECAI. Peer reviewed paper. Source record

[17] Wadden, D., Wennberg, U., Luan, Y., and Hajishirzi, H. (2019). Entity, Relation, and Event Extraction with Contextualized Span Representations. EMNLP IJCNLP. Peer reviewed paper. Source record

[18] Yao, Y., et al. (2019). DocRED: A Large-Scale Document-Level Relation Extraction Dataset. ACL. Peer reviewed paper. Source record

[19] Huguet Cabot, P. L., and Navigli, R. (2021). REBEL: Relation Extraction By End-to-end Language generation. Findings of EMNLP, 2370–2381. Peer reviewed paper. Source record

[20] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. ACL IJCNLP. Peer reviewed paper. Source record

[21] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. NeurIPS. Peer reviewed paper. Source record

[22] Yang, B., Yih, W. T., He, X., Gao, J., and Deng, L. (2015). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. ICLR. Conference paper with accessible preprint. Source record

[23] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. (2016). Complex Embeddings for Simple Link Prediction. ICML, PMLR 48, 2071–2080. Peer reviewed paper. Source record

[24] Sun, Z., Deng, Z. H., Nie, J. Y., and Tang, J. (2019). RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. ICLR. Peer reviewed paper. Source record

[25] Schlichtkrull, M., et al. (2018). Modeling Relational Data with Graph Convolutional Networks. ESWC. Conference paper with accessible preprint. Source record

[26] Zhu, Z., Zhang, Z., Xhonneux, L. P., and Tang, J. (2021). Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS. Peer reviewed paper. Source record

[27] Lajus, J., Galárraga, L., and Suchanek, F. (2020). Fast and Exact Rule Mining with AMIE 3. ESWC. Peer reviewed paper. Source record

[28] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Stanford technical report 1999-66. Technical report; legacy institutional record. Source record

[29] Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports, 9, 5233. Peer reviewed paper. Source record

[30] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. ICML, PMLR 70, 1321–1330. Peer reviewed paper. Source record

[31] Vrandečić, D., and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. Communications of the ACM, 57(10), 78–85. Published research article. Source record

[32] Himmelstein, D. S., et al. (2017). Systematic integration of biomedical knowledge prioritizes drugs for repurposing. eLife, 6, e26726. Peer reviewed paper. Source record

[33] Chandak, P., Huang, K., and Zitnik, M. (2023). Building a knowledge graph to enable precision medicine. Scientific Data, 10, 67. Peer reviewed data descriptor. Source record

[34] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS. Conference paper with accessible preprint. Source record

[35] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2024). Unifying Large Language Models and Knowledge Graphs: A Roadmap. IEEE Transactions on Knowledge and Data Engineering, 36(7), 3580–3599. Peer reviewed survey. Source record

[36] Edge, D., et al. (2024). From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv:2404.16130. Preprint; cited research version. Source record

[37] Sun, J., et al. (2024). Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. ICLR. Conference paper with accessible preprint. Source record

[38] Luo, L., Li, Y. F., Haffari, G., and Pan, S. (2024). Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. ICLR. Peer reviewed paper. Source record

[39] He, X., et al. (2024). G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. NeurIPS. Conference paper with accessible preprint. Source record

[40] Yao, S., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. ICLR. Conference paper with accessible preprint. Source record

[41] Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. (2025). LightRAG: Simple and Fast Retrieval-Augmented Generation. Findings of EMNLP, 10746–10761. Peer reviewed paper. Source record

[42] Cao, Y., Gao, Z., Li, Z., Xie, X., Zhou, S. K., and Xu, J. (2025). LEGO-GraphRAG: Modularizing Graph-Based Retrieval-Augmented Generation for Design Space Exploration. Proceedings of the VLDB Endowment, 18(10). Peer reviewed paper. Source record

[43] Dong, J., et al. (2025). Youtu-GraphRAG: Vertically Unified Agents for Graph Retrieval-Augmented Complex Reasoning. arXiv:2508.19855. Preprint; findings not independently reproduced here. Source record

[44] Ma, S., et al. (2025; first posted 2024). Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. arXiv:2407.10805, version 7. Preprint; cited research version. Source record

[45] Neo4j (version 5 documentation). Constraints. Cypher Manual. Official software documentation. Source record

[46] Neo4j (version 5 documentation). MATCH. Cypher Manual. Official software documentation. Source record

[47] Wolfram Research. LinearSolve. Wolfram Language Documentation. Official computational documentation. Source record

[48] Wolfram Research. PageRankCentrality. Wolfram Language Documentation. Official computational documentation. Source record

[49] Wilkinson, M. D., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data, 3, 160018. Peer reviewed principles paper. Source record

[50] Pearl, J. (2009). Causal inference in statistics: An overview. Statistics Surveys, 3, 96–146. Peer reviewed overview. Source record

[51] W3C (2014). RDF 1.1 Primer. W3C Working Group Note. Source record

[52] W3C (2013). SPARQL 1.1 Entailment Regimes. W3C Recommendation. Source record