

Secure Knowledge Graph Construction And Privacy-Preserving Question Answering For Industrial Knowledge Service-Oriented Distributed Digital Libraries

Jin Li*

Changsha Commerce & Tourism College, Changsha, 410116, China

* Corresponding author. E-mail: 18973155403@163.com

Received: Jul. 21, 2026; Accepted: Aug. 14, 2026

Industrial knowledge service-oriented distributed digital libraries integrate patents, standards, drawings, reports, and bibliographic records across nodes, facing collection fusion, entity redundancy, privacy leakage, and limited QA scalability. This study proposes a secure knowledge graph construction and privacy-preserving QA method combining metadata standardization, entity extraction, relation recognition, cross-library alignment, credibility-driven triple verification, graph partitioning, hotspot caching, parallel retrieval, and differential privacy ranking. Evaluated at five scales, it reaches 94.2%/92.6%/94.5%/93.1% entity F1/relation F1/alignment accuracy/triple validity, and QA accuracy/F1/MRR/Hit@5 of 93.6%/92.1%/91.3%/95.2%. At 64 nodes, latency is 360 ms (vs. 950/1650 ms for Federated/Centralized KG-QA), with 1650 queries/s at 10,000 concurrent queries.

Keywords: Industrial knowledge service, distributed digital library, secure knowledge graph, privacy-preserving question answering, entity alignment, differential privacy

© The Author('s). This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202701_36.013

1. Introduction

Digital libraries are evolving into distributed, industrial knowledge service systems, with data scattered across nodes under inconsistent metadata and permissions. Keyword retrieval struggles with deep semantic relationships, while centralized QA requires aggregating raw data — including sensitive documents — risking compromised autonomy and bottlenecks [1, 2]. A secure, scalable knowledge graph protecting privacy is thus key for such digital libraries.

Existing knowledge graph research offers many entity extraction, relation, alignment, and reasoning methods [3–5], and KG-QA improves interpretability via path reasoning [6–8], but mostly assumes centralized storage. Privacy-preserving retrieval and federated learning reduce leakage risk [9, 10] but still face query-intent exposure and mali-

cious injection; accuracy, privacy, robustness, and scalability must be evaluated together.

Unlike centralized KG-QA or federated retrieval, the framework's novelty is threefold: entity alignment jointly constrained by semantic similarity, secure summaries, and security tags; credibility-driven triple verification before graph fusion; and privacy-preserving ranking integrating anonymization, differential privacy, path credibility, and latency. Section 2 covers methods, Section 3 results, Section 4 conclusions.

Distributed digital libraries are large-scale systems whose core challenges include heterogeneous resource integration, cross-node retrieval, concurrent access, access control, and dynamic updates. Multi-library collaboration must support unified retrieval without over-centralizing raw data [11], making distributed storage, graph parti-

tioning, parallel querying, and incremental updates the technical foundation for intelligent digital libraries.

Industrial knowledge service platforms, which must integrate patents, technical standards, engineering drawings, and enterprise technical reports alongside conventional bibliographic resources, exemplify these challenges and place additional demands on cross-node security and credibility control.

Knowledge graphs represent documents, authors, institutions, keywords, citations, and locations as structured networks, shifting libraries from “retrieval” to “discovery.” Entity alignment and knowledge embedding continue developing [12], but cross-library scenarios face name variants and duplicate records; without reliable alignment, graphs suffer redundancy and inconsistent answers.

KG-QA transforms questions into structured graph reasoning via entity recognition, relation matching, path search, and ranking, returning more interpretable paths than keyword retrieval. Complex questions need multi-hop reasoning, and latency grows with graph size and node distribution, so scalable QA must weigh throughput, latency, and memory alongside accuracy.

Privacy protection is unavoidable in distributed digital library QA: continuous queries may expose research interests, and cross-node alignment may leak local metadata. Differential privacy, federated learning, and anonymization mitigate these risks [13–15], but often introduce noise or reduce ranking stability, requiring joint optimization of accuracy, reliability, privacy, and latency rather than single-metric evaluation.

2. Methods

2.1. System Model and Problem Definition

2.1.1. Distributed Digital Library System Architecture

The distributed digital library comprises collection nodes, an entity alignment service, a secure knowledge graph service, a privacy-preserving QA service, and a user interface (Fig. 1). Each node keeps data locally, sharing only anonymized candidates with the collaboration layer. A query is intent-identified, desensitized, entity-linked, path-retrieved, ranked, and answered with source credibility and path explanation.

$$G_d = (N, E, R, S), \quad n_i \in N \quad (1)$$

where N represents the set of collection nodes participating in the collaboration; E represents the set of cross-library entities, including documents, authors, institutions, topics, and collection nodes; R represents the set of semantic relationships between entities; and S represents the set of secu-

rity tags used to identify states such as public, restricted, sensitive, and pending verification.

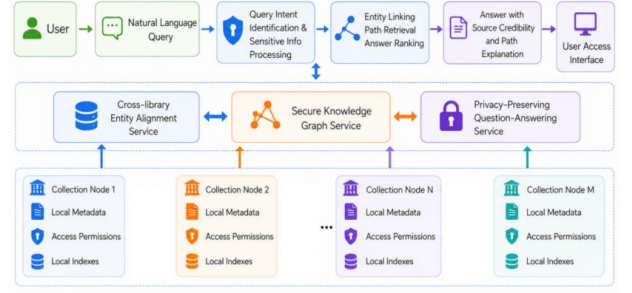


Fig. 1. Overall system architecture of the proposed industrial knowledge service-oriented distributed digital library framework

2.1.2. Node-Level Dataset and Knowledge Unit Definition

The dataset for each collection node is defined as follows:

$$G_i = (V_i, E_i, X_i, L_i), \quad V = \bigcup_{i=1}^M V_i, \quad E = \bigcup_{i=1}^M E_i \quad (2)$$

where G_i represents the local knowledge graph formed by the i collection node; V_i represents the set of entities within that node; E_i represents the set of relation edges within that node; and G represents the fused global security knowledge graph. This formula connects the local graph of a single library to the global graph across multiple libraries.

$$D_i = \{B_i, P_i, A_i, O_i, K_i, C_i, L_i\} \quad (3)$$

where B_i represents the set of book resources in the i collection node; P_i represents the set of papers and conference literature resources; A_i represents the set of author entities; O_i represents the set of institutional entities; K_i represents the set of keywords and subject terms; C_i represents the set of citation relationships; and L_i represents the set of collection location and access status.

In industrial knowledge service scenarios, this resource set can be further extended to include patents, technical standards, and engineering drawings, which are modelled using the same entity, relationship, attribute, and security-tag structure described below.

Knowledge units are divided into entities (documents, authors, institutions, topics, journals, nodes), relationships (writing, publication, affiliation, citation, same-author variants, collection location), attributes (year, language, classification, DOI, access level), and security tags (public, restricted, sensitive, pending), which constrain merging, path search, and ranking rather than ordinary similarity calculations.

2.1.3. Security Threat Model

Four security threats are considered: entity alignment may leak local metadata; continuous query logs may be re-identified; malicious nodes may inject incorrect triples; and reasoning paths may indirectly reveal sensitive topics. Accuracy, credibility, privacy risk, and latency are thus unified into one objective function.

Access-control constraints are embedded across the knowledge unit lifecycle, with security labels and logs retained from extraction through QA invocation. Restricted collections may aid topic statistics and disambiguation but not full-field exposure in answers.

$$\text{Risk}(q) = \sigma(\eta_1 I_q + \eta_2 H_q + \eta_3 C_q + \eta_4 A_q) \quad (4)$$

where I_q represents the identifiability of the query q ; H_q represents the strength of the association between the query and historical sessions; C_q represents the intensity of sensitive topics; and A_q represents the degree of exposure of the answer field. All four variables are normalized to the interval [0,1], with larger values indicating higher privacy risks.

$$\max F = \alpha \text{Acc}(q, a) + \beta \text{Trust}(a) - \gamma \text{Risk}(q) - \delta \text{Lat}(q) \quad (5)$$

where $\text{Acc}(q, a)$ represents the semantic accuracy between question q and candidate answer a ; $\text{Trust}(a)$ represents the path credibility of the candidate answer; $\text{Risk}(q)$ represents query privacy risk; $\text{Lat}(q)$ represents query response latency; α, β, γ and δ are weight parameters used to balance accuracy, credibility, privacy protection, and system efficiency.

2.1.4. Query Types and Privacy Constraints

Queries are divided into ordinary, sensitive-topic, and continuous-behavior categories: ordinary queries prioritize relevance, sensitive-topic queries need stronger desensitization, and continuous-behavior queries need re-identification control — e.g., repeated searches for specific diseases or unpublished materials should trigger stronger protection.

$$\Omega(q) = \begin{cases} 0, & q \in Q_{\text{normal}} \\ 1, & q \in Q_{\text{sensitive}} \\ 2, & q \in Q_{\text{session}} \end{cases} \quad (6)$$

where T_q represents the privacy type of query q ; $T_q = 0$ represents a normal query, $T_q = 1$ represents a sensitive topic query, and $T_q = 2$ represents a continuous behavior query. This hierarchical variable was used to control the subsequent desensitization intensity, privacy budget, and answer granularity.

2.1.5. System Optimization Objective Modelling

The system optimization objective jointly considers accuracy, credibility, privacy risk, and latency — a multi-objective constrained optimization rather than simple accuracy maximization.

2.1.6. Source Reliability and Dynamic Quality Control

Data reliability varies across nodes — national centers, publishers, journals, patent offices, and local catalogues differ in authority. A reliability prior is assigned per source node and adjusted dynamically during triple verification, rising when evidence agrees with reliable nodes and falling when conflicts recur, giving graph construction adaptive quality control.

2.2. Secure Knowledge Graph Construction Methods

2.2.1. Multi-Source Metadata Cleaning and Standardization

Fig. 2 shows the construction workflow: the system receives MARC, Dublin Core, paper metadata, thesaurus, citation, and location data from different nodes, standardizes fields and names, then extracts entities (documents, authors, institutions, topics, keywords, publications, nodes) and relations (authored_by, published_in, belongs_to, cites, has_subject, stored_in, similar_to).

$$x_{i,k}^{std} = \phi_k(x_{i,k}), \quad r_{i,k}^{std} = \psi_k(r_{i,k}) \quad (7)$$

where the original attribute value of each field in a collection node is mapped by a normalized mapping function to a standardized unified attribute representation, for format unification, semantic normalization, and noise correction; and each original relation field is likewise mapped by a semantic mapping function to a normalized standard relation representation, unifying relation naming, direction constraints, and semantic expression.

2.2.2. Entity Alignment and Merging Decision

After metadata standardization, entities originating from different collection nodes that plausibly refer to the same real-world object must be merged into a single global identity. The merging decision jointly considers the overall similarity score and the consistency of key evidence, and is formulated as follows:

$$z_{ij} = \begin{cases} 1, & S(e_i, e_j) \geq \tau_a \text{ and } C(e_i, e_j) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where z_{ij} represents the merging decision between entities e_i and e_j , $z_{ij} = 1$ indicates merging, and $z_{ij} = 0$ indicates not merging; τ_a represents the entity alignment threshold; $C(e_i, e_j)$ represents the key evidence consistency

constraint, which is used to determine whether strong evidence such as DOI, ISBN, institutional context, co-authors or citation neighborhood conflict.

$$S(e_i, e_j) = \lambda_1 S_n + \lambda_2 S_s + \lambda_3 S_c + \lambda_4 S_r \quad (9)$$

where S_n represents name string similarity; S_s represents semantic embedding similarity; S_c represents contextual neighborhood similarity; S_r represents reference neighborhood or co-occurrence structure similarity; λ_1 to λ_4 represent the weights of different similarity components, satisfying $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$. The system merges entities only when the overall similarity exceeds the threshold and the key contextual evidence is consistent with this.

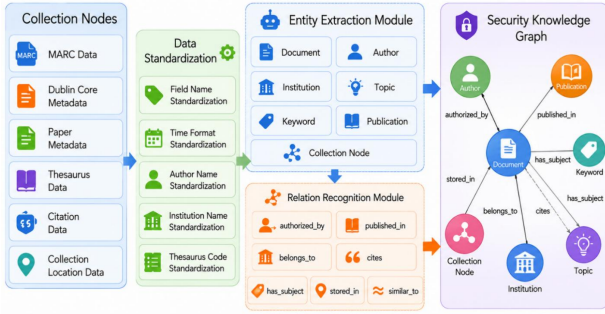


Fig. 2. Secure knowledge graph construction workflow

2.2.3. Credibility-Driven Triple Verification

To avoid erroneous knowledge entering the graph, this study designs a credibility-driven triple verification mechanism. Each triple (h, r, t) calculates a credibility score from four dimensions: source reliability, cross-library consistency, semantic confidence, and time freshness.

$$T_{tri}(t) = \rho_1 R_s(t) + \rho_2 R_c(t) + \rho_3 R_m(t) + \rho_4 R_t(t) \quad (10)$$

where $T(h, r, t)$ represents the overall credibility of the triple (h, r, t) ; h is the head entity, r is the relation, and t is the tail entity; Q_s represents source reliability, Q_c represents cross-library consistency, Q_m represents extraction model confidence, and Q_t represents time freshness. This formula evaluates the credibility of the triple from four aspects: source reliability, cross-library consistency, model confidence, and time freshness.

$$T(h, r, t) = \mu_1 Q_s + \mu_2 Q_c + \mu_3 Q_m + \mu_4 Q_t \quad (11)$$

where Q_s represents the reliability of the source node; Q_c represents the consistency of support for the same triple

from multiple collection nodes; Q_m represents the confidence level of the entity extraction and relation recognition model output; Q_t represents the time freshness of the triple; and μ_1 to μ_4 are confidence weights. Low-confidence triples do not directly enter the core graph but instead enter the verification queue.

2.2.4. Candidate Blocking and Fine-Grained Matching

Cross-library entity alignment uses two stages: candidate blocking, which quickly narrows candidates using normalized names, DOIs, ISBNs, subject codes, and institutional phrases; and fine-grained matching, integrating name, semantic-vector, contextual-neighborhood, and citation-structure similarity. High-risk entities require at least two independent evidence sources to meet the threshold simultaneously, avoiding incorrect merges from single-field errors.

$$B(e) = \text{hash}(\text{norm}(\text{name}(e))) \oplus \text{hash}(\text{id}(e)) \oplus \text{hash}(\text{topic}(e)) \quad (12)$$

where $B(e)$ represents the candidate blocking set corresponding to entity e ; $\text{key}(e)$ represents the blocking key generated by normalized name, DOI, ISBN, subject term encoding, and institutional phrase; $\Omega(\text{key}(e))$ represents the candidate entity index associated with the blocking key. The candidate blocking function is used to quickly narrow down the candidate entity set, thereby reducing the computational overhead of pairwise comparisons across libraries.

Entity extraction combines dictionary rules, context encoding, and confidence filtering to resolve author-name and subject-term variants. Surface forms, context vectors, and co-citation relationships feed candidate generation, which alignment resolves into a single object [2, 3], reducing “same name, different people” errors. Field mapping unifies MARC, Dublin Core, and paper metadata formats; missing fields use contextual completion and rule constraints.

2.3. Privacy-Preserving Question Answering Model

2.3.1. Query Intent Recognition and Question Type Determination

The privacy-preserving QA model has five stages: query intent identification, sensitive-information de-identification, candidate path retrieval, differential privacy ranking, and secure answer generation. Questions are categorized by type (document search, author association, topic expansion, citation tracking, collection location, comprehensive reasoning), each mapping to different entity-linking strategies and path-search depths.

$$P(c | q) = \text{softmax}(W_c h_q + b_c) \quad (13)$$

where h_q represents the semantic encoding vector of question q ; W_c and b_c represent the weight matrix and bias term of the question type classifier, respectively; c represents the candidate question type; $P(c | q)$ represents the probability that question q belongs to type c , and the output probability is used to select subsequent entity links and path search strategies.

2.3.2. Sensitive Information Desensitization and Query Anonymization

Query anonymization processes user identity, query time, session identifiers, and sensitive keywords, retaining only the minimum information needed to answer while desensitizing high-risk topics. For restricted collections, sensitive topics, and continuous sessions, the system prioritizes generalized replacement, session-identifier reset, and minimum-necessary-feature retention so subsequent graph retrieval never directly exposes the full query text.

$$\tilde{q} = \text{Mask}(q, U, T, S), \quad I(\tilde{q}; u) \leq \kappa \quad (14)$$

where $\tilde{q}$ represents the query after anonymization and desensitization; u represents the user identity or session identifier; $I(\tilde{q}; u)$ represents the mutual information between the anonymized query and the user identity; and κ represents the upper limit of allowed information exposure. This formula indicates that the information exposure between the anonymized query and user identity is constrained by a threshold.

2.3.3. Graph Path Reasoning and Candidate Answer Generation

The graph path reasoning stage converts the question into entity nodes and relational constraints, retrieving multi-hop candidate paths in the knowledge graph. Candidate paths include not only the starting entity and the target answer but also evidence nodes such as intermediate topics, authors, institutions, journals, citation relationships, and library locations, thus making the question-and-answer results interpretable.

$$\text{Path}(q) = \{v_0, r_1, v_1, \dots, r_k, v_k\} \quad (15)$$

where v_0 represents the starting entity node linked from the user's question; v_l represents the entity node reached by the l hop; r_l represents the relation edge used by the l hop; and k represents the maximum number of hops in the path. Candidate paths are used not only to generate answers but also to explain their source.

$$\begin{aligned} \text{Score}(a) = & \eta_1 \text{Rel}(q, a) + \eta_2 \text{Trust}(p) - \eta_3 \text{Risk}(q, a) \\ & - \eta_4 \text{Lat}(p) + \text{Lap}(\Delta f / \epsilon) \end{aligned} \quad (16)$$

2.3.4. Secure Answer Ranking under Differential Privacy Constraints

Where $\text{Rel}(q, a)$ represents the semantic relevance between question q and candidate answer a ; $\text{Trust}(p)$ represents the credibility of candidate path p ; $\text{Risk}(q, a)$ represents the privacy exposure risk caused by answer a to query q ; $\text{Lat}(p)$ represents the retrieval delay of path p ; and $\text{Lap}(\Delta f / \epsilon)$ represents the Laplace perturbation determined by the sensitivity Δf of the scoring function and the privacy budget ϵ .

To avoid unavailability from over-protection, this study uses a tiered privacy budget: weaker perturbation for public topics and general location queries preserves retrieval efficiency and accuracy, while stronger perturbation for personal-interest, sensitive-research, or restricted-collection queries limits fine-grained identity clues in returned results [1, 15], dynamically balancing service quality and privacy across security levels.

Answer ranking jointly scores relevance, credibility, privacy risk, and latency: relevance comes from the query vector and candidate-path semantics; credibility from triple credibility and source-node reliability; privacy risk from sensitive-topic intensity, query-sequence identifiability, and answer exposure; and the latency term constrains overhead from complex path search — ensuring results are accurate, explainable, and privacy-controlled.

$$\text{Lap}(\Delta f / \epsilon) \sim \frac{\epsilon}{2\Delta f} \exp\left(-\frac{\epsilon|x|}{\Delta f}\right) \quad (17)$$

where ϵ represents the privacy budget, with smaller values indicating stronger privacy protection; Δf represents the global sensitivity of the answer scoring function; $\text{Lap}(\Delta f / \epsilon)$ represents Laplace noise scaled to $\Delta f / \epsilon$; and the perturbation term is used to reduce the risk of the ranking output revealing user preferences.

2.3.5. Question Answering Explanation and Parameter Update

The QA explanation module outputs the evidence path, source-node count, path credibility, and privacy-processing status alongside the final answer — e.g., for a question on privacy-preserving knowledge graphs in digital libraries, the system shows a multi-hop evidence chain of topic, paper, author, journal, and citation nodes, letting users judge source reliability.

$$\begin{aligned} \text{Score}(p | q) = & \mu_1 \text{Rel}(q, p) + \mu_2 \text{Trust}(p) \\ & - \mu_3 \text{Risk}(p) - \mu_4 |p| \end{aligned} \quad (18)$$

where $\text{Score}(p | q)$ represents the comprehensive score of candidate path p given question q ; $\text{Rel}(q, p)$ represents the semantic relevance between the question and the path; $\text{Trust}(p)$ represents the aggregated result of the credibility

of triples on the path; $\text{Risk}(p)$ represents the risk of sensitive information exposure that the path may cause; $|p|$ represents the path length; and η_1 to η_4 are weight parameters. The candidate path score simultaneously considers relevance, path credibility, privacy risk, and path length.

The privacy-preserving QA model follows an “identification-de-identification-retrieval-ranking-interpretation” pipeline. After query input, the model identifies core entities, keywords, time range, and question type, then checks for sensitive topics, personal-interest trajectories, or restricted-collection clues; for sensitive segments, it applies generalized replacement, session-identifier reset, and minimum-necessary-feature retention so subsequent retrieval never exposes the full query text.

The interpretation module outputs the evidence path, source-node count, and credibility score alongside the answer, improving verifiability for users and helping maintainers locate erroneous entities, low-trust relationships, and privacy-risk sources. When an answer is downgraded for privacy reasons, internal audit logs are retained without exposing perturbation values or sensitive judgment rules to ordinary users.

2.4. Distributed Scalable Implementation Mechanism

2.4.1. Hybrid Graph Partitioning and Distributed Storage

To accommodate large-scale collections and high-concurrency queries, a hybrid graph-partitioning mechanism jointly partitions by subject domain, entity type, node, and citation community, keeping related topics or high-frequency paths adjacent. Popular authors, journals, and keywords get a hot-entity cache, while complex QA (entity recognition, path retrieval, credibility calculation, ranking) executes in parallel across computing units.

$$\min_{\Pi} \sum_{(u,v) \in E} I[\Pi(u) \neq \Pi(v)] w_{uv} + \rho \sum_k |V_k - \bar{V}| \quad (19)$$

where P_m represents the m graph partition; $\text{cut}(P_m, P_n)$ represents the number of cross-partition edges between partitions P_m and P_n ; $\text{load}(P_m)$ represents the computational and storage load of partition P_m ; L_{avg} represents the average load; and ρ represents the load balancing penalty coefficient. The graph partitioning objective controls both the number of cross-partition edges and the load balancing of each partition.

2.4.2. Secure Digest Exchange and Communication Optimization

Cross-node communication uses candidate-digest exchange rather than complete original records — exchanging encrypted digests, local vector indexes, and trusted-

evidence statistics to reduce network load and metadata-leakage risk. A query audit module detects abnormally high-frequency queries, continuous sensitive-topic queries, and abnormal cross-node access, increasing privacy perturbation or restricting return-path granularity when re-identification or probing is detected.

$$\text{Comm} = \sum_{i=1}^M \sum_{j \neq i} |H_{ij}| \cdot d_h \quad (20)$$

where H_{ij} represents the set of candidate summaries exchanged between node i and node j ; $|H_{ij}|$ represents the number of summaries; d_h represents the dimension of the summary vector; and $\text{Comm}(i, j)$ represents the communication overhead between nodes.

2.4.3. Computational Complexity and System Overhead Analysis

Entity-extraction complexity scales roughly linearly with local record count; entity-alignment complexity is reduced via candidate blocking and vector nearest-neighbor retrieval; path-query complexity depends on partitioning quality, path depth, and candidate count; and privacy-perturbation/ranking overhead is relatively small. Overall, partitioning, caching, parallelism, and incremental updates split the centralized bottleneck across nodes, suiting scalable information-system scenarios.

$$T_{\text{total}} = O(n) + O(m \log m) + O(b^k d) + O(c \log c) \quad (21)$$

where n represents the number of local records; m represents the number of candidate entities; d represents the vector dimension; k represents the path search depth; b represents the average branching factor; and c represents the number of candidate answers. This complexity consists of local extraction, candidate nearest-neighbor retrieval, path search, and answer ranking.

2.4.4. Incremental Update and Consistency Maintenance

Cross-node communication exchanges encrypted digests and vector indexes rather than complete records, reducing network load and leakage risk. A query audit module detects abnormal query frequency or cross-node access, increasing perturbation or restricting path granularity when probing is detected.

$$\Delta G^t = (\Delta V^t, \Delta E^t), \quad G^t = G^{t-1} \oplus \Delta G^t \quad (22)$$

where G^t represents the knowledge graph at time t ; ΔV represents the newly added entity set; ΔE represents the newly added relation set; $U(\cdot)$ represents the local update operator; and G^{t+1} represents the updated knowledge

graph. This formula indicates that the system only performs local merging and index updates on newly added entities and relations, avoiding a full reconstruction.

2.5. Experimental Design

2.5.1. Experimental Environment and Dataset Construction

A semi-synthetic distributed digital library environment was built with 4-64 collection nodes from public bibliographic and metadata formats, reproducing cross-library heterogeneity while avoiding real behavior disclosure. Data include books, articles, patents, standards, drawings, reports, authors, institutions, keywords, citations, and anonymous query logs across five scales (Table 1).

Table 1. Simulated industrial knowledge service-oriented distributed digital library datasets

Scale	Library Nodes	Entities	Relations	Queries
Small	4	50,000	180,000	5,000
Medium	8	120,000	460,000	10,000
Large	16	300,000	1,200,000	20,000
Extra-Large	32	800,000	3,600,000	50,000
Ultra-Large	64	1,500,000	7,200,000	100,000

2.5.2. Data Scale and Simulation Settings

The incremental update mechanism handles continuous library growth: new records are locally extracted and desensitized before the collaboration layer updates only affected entities and paths, avoiding full reconstruction; high-impact changes use batch window updates.

The query set covers six task types — document search, author association, topic expansion, citation tracking, collection location, and reasoning — each at three security levels (public, restricted, sensitive), simple queries testing entity linking and complex queries requiring multi-hop reasoning. Recorded metrics include answer accuracy, path reliability, privacy leakage risk, query latency, and cross-node communication count.

2.5.3. Comparison Methods and Evaluation Metrics

Comparison methods included BM25, BERT-QA, KG-QA, GraphSAGE-KGQA, Centralized/Federated KG-QA, and the proposed method, evaluated on QA performance, graph quality, system performance, privacy/security, and dynamic maintenance metrics (Table 2).

The entity alignment candidate threshold was set to 0.72, the minimum threshold for trusted triples entering the core graph was set to 0.68, the maximum path search depth was 3 hops, the hotspot cache capacity was 15% of the high-frequency entity set, and the concurrent query load was set to 500, 1000, 2000, 5000, and 10000. The privacy budget was set to 0.1, 0.5, 1.0, 2.0, and 5.0 to observe the relationship between accuracy and leakage risk.

The five data scales simulate 4, 8, 16, 32, and 64 nodes, each with bibliographic, paper, patent/standard records, author/institution information, keywords, citations, and collection-location fields, with three injected noise types — name-abbreviation/translation variants, institution-hierarchy inconsistencies, and keyword synonyms — to approximate a real industrial knowledge service scenario. The query set covers the same six question types described above.

3. Experimental results and discussion

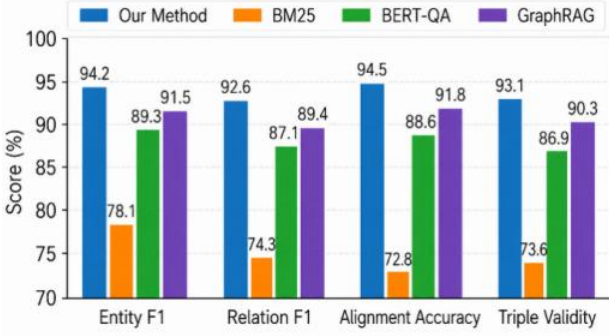
This section reports the experimental results obtained across the five simulated data scales described in Section 2.5, covering knowledge graph construction quality, question answering performance, system scalability, privacy protection, and robustness against security attacks, and concludes with a discussion of their practical implications for industrial knowledge service-oriented distributed digital libraries.

3.1. Knowledge Graph Construction Quality Comparison

Fig. 3 compares knowledge graph construction quality across KG-QA, GraphSAGE-KGQA, Centralized KG-QA, Federated KG-QA, and the proposed method. The proposed method achieves the highest entity F1 (94.2%), relation F1 (92.6%), alignment accuracy (94.5%), and triple validity (93.1%), benefiting from the joint similarity-evidence entity alignment described in Section 2.2.2 and the credibility-driven triple verification described in Section 2.2.3, which together suppress erroneous merges and low-confidence triples before they enter the graph. Centralized KG-QA attains comparable entity extraction quality but shows lower alignment accuracy and triple validity because it lacks cross-library credibility weighting, while Federated KG-QA's reliance on local-only training limits its ability to resolve cross-node name variants and duplicate records, resulting in more residual redundancy in the fused graph.

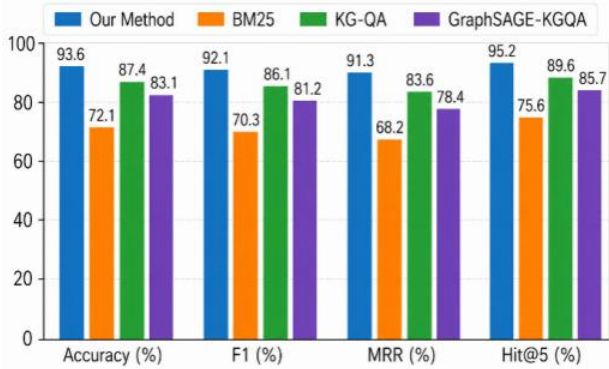
Table 2. Evaluation metrics used in the simulation

Metric Type	Indicators	Purpose
QA Performance	Accuracy, Precision, Recall, F1, MRR, Hit@5	Evaluate answer quality
KG Quality	Entity F1, Relation F1, Alignment Accuracy, Triple Validity	Evaluate graph construction
System Efficiency	Latency, Throughput, Construction Time, Memory Cost	Evaluate scalability
Privacy Security	Leakage Risk, Re-identification Rate, Attack Success Rate	Evaluate defence ability
Dynamic Updating	Update Time, Consistency Rate	Evaluate incremental maintenance

**Fig. 3.** Knowledge graph construction quality comparison

3.2. Question Answering Accuracy Comparison

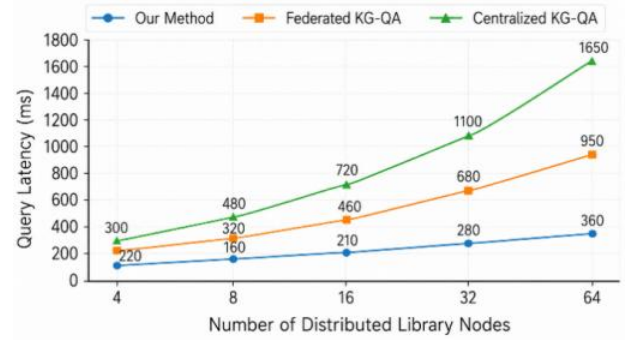
Fig. 4 compares QA performance: the proposed method leads on Accuracy, F1, MRR, and Hit@5, with the largest gains on multi-hop questions rather than simple search, since it combines semantic matching with path reasoning and path credibility in ranking. KG-QA loses stability with entity ambiguity; GraphSAGE-KGQA lacks privacy constraints and trusted filtering.

**Fig. 4.** Question answering performance comparison

3.3. Query Latency under Distributed Node Scale

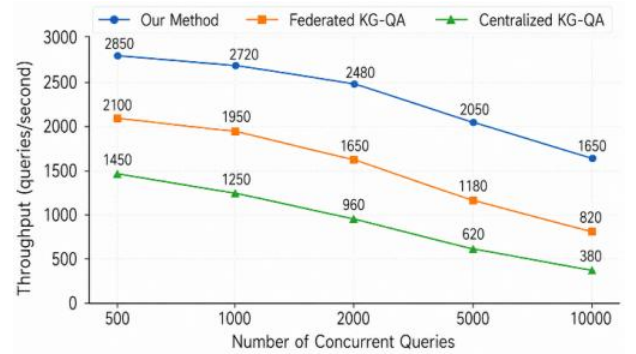
Fig. 5 shows query latency as nodes expand from 4 to 64: Centralized KG-QA's latency rises quickly as it must handle more cross-node entities and candidate paths; Federated KG-QA reduces centralization pressure but retains signifi-

cant communication overhead. The proposed method's use of graph partitioning, hotspot caching, and parallel path retrieval yields a far more gradual latency increase, suiting scalable distributed-library QA.

**Fig. 5.** Query latency under different numbers of distributed library nodes

3.4. System Throughput under Concurrent Query Load

Fig. 6 shows throughput under concurrent loads from 500 to 10,000: all methods' throughput declines or slows, but the proposed method maintains high processing capacity, since hot-entity caching reduces repeated path-search overhead and parallel processing reduces per-node wait time — supporting both accuracy and high-concurrency access.

**Fig. 6.** System throughput under different query loads

3.5. Construction Time with Graph Size Growth

Fig. 7 compares construction time as entities grow from 50,000 to 1,500,000: centralized construction time rises dramatically; the federated method helps somewhat but is still affected by cross-node alignment and communication overhead. The proposed method reduces redundant computation via candidate blocking, summary exchange, and incremental triple validation, achieving lower cost at large scale.

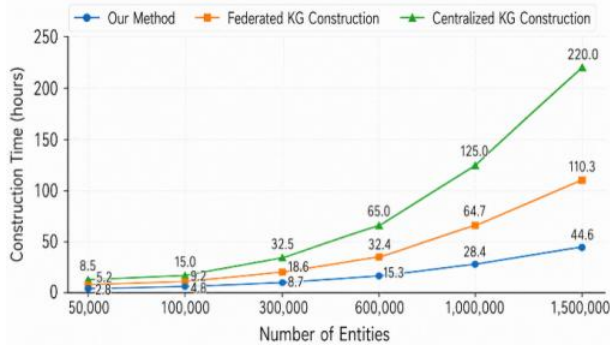


Fig. 7. Graph construction time under different graph scales

3.6. Memory Consumption with Graph Size Growth

Fig. 8 shows memory consumption: centralized construction requires maintaining all entities, edges, and indexes on one node, causing high memory pressure; Federated KG-QA distributes some computation but still needs significant synchronization state. The proposed method's partitioned storage and hotspot caching avoid expanding the full graph on a single node, keeping memory growth controllable.

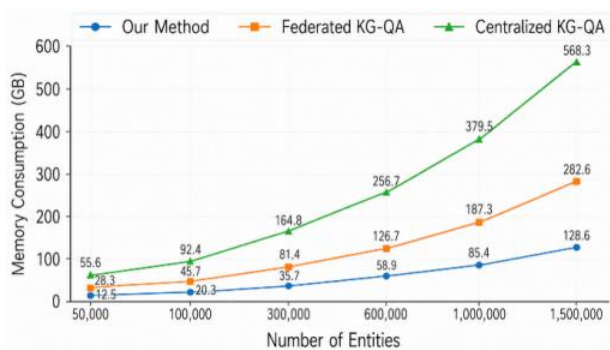


Fig. 8. Memory consumption under different graph scales

3.7. Impact of Privacy Budget on Experiment

Fig. 9 shows privacy-budget effects: at 0.1, perturbation is strong and leakage risk lowest, but answer ranking suffers; at 5.0, accuracy approaches the unperturbed state but

leakage risk rises sharply. This confirms the privacy budget should be dynamically adjusted by query type and resource sensitivity — larger for general collection-location queries, smaller for sensitive topics and continuous-behavior queries.

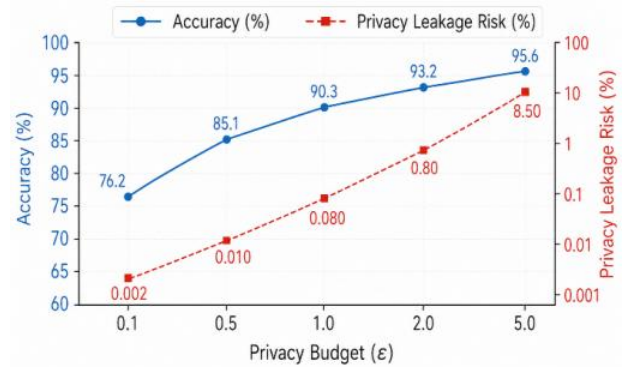


Fig. 9. Accuracy and privacy leakage risk under different privacy budgets

3.8. Robustness Experiments Against Security Attacks

Fig. 10 shows robustness against member inference, re-identification, and malicious injection: the proposed method achieves low success rates for all three, since anonymization reduces linkability, differential privacy limits ranking exposure, and credibility verification blocks malicious triples — showing privacy-preserving QA needs combined defenses rather than single-point anonymization.

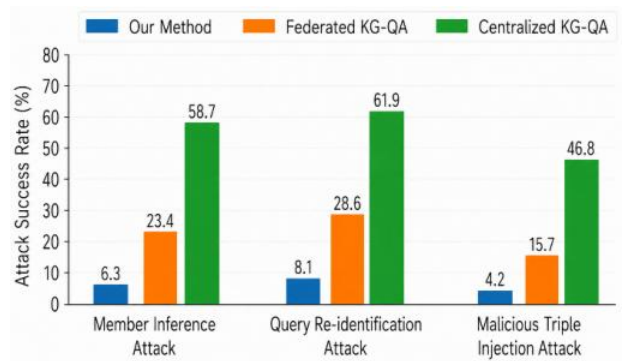


Fig. 10. Robustness comparison under privacy attacks and malicious injection attacks

New entities concentrate on popular topics rather than arriving evenly, risking hot-topic congestion. This study injects topic-popularity skew so 20% of topics bear about 60% of query load; results show hot-topic caching and partitioning prevent response degradation under load. Document-

search accuracy depends on entity-link accuracy, collection-location on cache hit rate, and topic-expansion/citation-tracking on multi-hop path quality — gains are largest on complex, cross-node reasoning.

3.9. Discussion on Applicability of the Method

Unlike ordinary library or enterprise document management systems, this method focuses on knowledge organization, secure integration, and privacy-preserving QA for industrial knowledge service-oriented systems combining bibliographic resources with patents, standards, drawings, and reports. The design considers multi-node collaboration, graph growth, concurrent queries, privacy attacks, and updates from the outset.

3.10. Discussion on Knowledge Graph Quality and Question Answering Performance

Results show a secure knowledge graph improves QA accuracy mainly because structured paths compensate for keyword retrieval and simple semantic matching, while credibility verification reduces the impact of erroneous merges and low-quality triples. For complex queries, the system returns not just the answer but source node, relation path, and credibility explanation, better suiting academic retrieval in digital libraries.

3.11. Discussion on Privacy Protection and System Overhead

Privacy protection introduces controllable accuracy loss: differential privacy limits ranking exposure, anonymization reduces re-identification risk, and the privacy budget can be adjusted by user type and sensitivity. This could extend to an interactive interface showing the knowledge path and credibility scores; combined with a future LLM, generative answers should strictly rely on credible graph evidence to avoid fabricated content.

4. Conclusions

This study proposes a secure knowledge graph construction and privacy-preserving question-answering method for industrial knowledge service-oriented distributed digital libraries. The main conclusions are as follows:

- A secure knowledge graph construction framework was developed that integrates multi-source metadata standardization, entity extraction, relation recognition, cross-library entity alignment, and credibility-driven triple verification, enabling reliable knowledge fusion across heterogeneous industrial knowledge service nodes.

- A privacy-preserving question-answering model combining query anonymization, differential privacy constraints, graph path reasoning, and trusted answer ranking was proposed, reducing the risk of user query preference leakage while maintaining answer quality.
- Experimental results show that the proposed method achieves a 94.2% F1 score for entity recognition, 92.6% for relation recognition, 94.5% alignment accuracy, and 93.1% triple validity in knowledge graph construction, and 93.6% accuracy, 92.1% F1, 91.3% MRR, and 95.2% Hit@5 in question answering, outperforming comparison methods including BM25, BERT-QA, KG-QA, GraphSAGE-KGQA, Centralized KG-QA, and Federated KG-QA.
- The proposed method maintains low query latency (360 ms at 64 distributed nodes) and high throughput (1650 queries/s under 10,000 concurrent queries), and shows strong robustness against member inference, query re-identification, and malicious triple injection attacks.
- The framework is suitable for secure knowledge organization, privacy-preserving retrieval, and scalable intelligent question-answering services in large-scale industrial knowledge service-oriented distributed digital libraries.
- Future work will focus on validating the framework on real multi-institution and multi-enterprise datasets, constructing reproducible public benchmarks, exploring secure large-language-model-based KG-QA, and introducing finer-grained access control and audit mechanisms.

References

- [1] C. Chen, F. Zheng, J. Cui, Y. Cao, G. Liu, J. Wu, and J. Zhou, (2024) “Survey and Open Problems in Privacy-Preserving Knowledge Graph: Merging, Query, Representation, Completion, and Applications” **International Journal of Machine Learning and Cybernetics** 15(8): 3513–3532. DOI: [10.1007/s13042-024-02106-6](https://doi.org/10.1007/s13042-024-02106-6).
- [2] N. Fanourakis, V. Efthymiou, D. Kotzinos, and V. Christophides, (2023) “Knowledge Graph Embedding Methods for Entity Alignment: An Experimental Review” **Data Mining and Knowledge Discovery** 37(5): 2070–2137. DOI: [10.1007/s10618-023-00941-9](https://doi.org/10.1007/s10618-023-00941-9).

- [3] B. D. Trisedya, F. D. Salim, J. Chan, D. Spina, F. Scholer, and M. Sanderson, (2023) "i-Align: An Interpretable Knowledge Graph Alignment Model" **Data Mining and Knowledge Discovery** 37(6): 2494–2516. DOI: [10.1007/s10618-023-00963-3](https://doi.org/10.1007/s10618-023-00963-3).
- [4] N. Ibrahim, S. Aboulela, A. Ibrahim, and R. Kashef, (2024) "A Survey on Augmenting Knowledge Graphs (KGs) with Large Language Models (LLMs): Models, Evaluation Metrics, Benchmarks, and Challenges" **Discover Artificial Intelligence** 4: DOI: [10.1007/s44163-024-00175-8](https://doi.org/10.1007/s44163-024-00175-8).
- [5] X. Zhu, Z. Li, X. Wang, X. Jiang, P. Sun, X. Wang, Y. Xiao, and N. J. Yuan, (2024) "Multi-Modal Knowledge Graph Construction and Application: A Survey" **IEEE Transactions on Knowledge and Data Engineering** 36(2): 715–735. DOI: [10.1109/TKDE.2022.3224228](https://doi.org/10.1109/TKDE.2022.3224228).
- [6] Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, and J.-R. Wen, (2023) "Complex Knowledge Base Question Answering: A Survey" **IEEE Transactions on Knowledge and Data Engineering** 35(11): 11196–11215. DOI: [10.1109/TKDE.2022.3223858](https://doi.org/10.1109/TKDE.2022.3223858).
- [7] S. Aghaei, E. Raad, and A. Fensel, (2022) "Question Answering over Knowledge Graphs: A Case Study in Tourism" **IEEE Access** 10: 69788–69801. DOI: [10.1109/ACCESS.2022.3187178](https://doi.org/10.1109/ACCESS.2022.3187178).
- [8] Y. Song, W. Li, G. Dai, and X. Shang, (2023) "Advancements in Complex Knowledge Graph Question Answering: A Survey" **Electronics** 12(21): DOI: [10.3390/electronics12214395](https://doi.org/10.3390/electronics12214395).
- [9] X. Chen, S. Jia, and Y. Xiang, (2020) "A Review: Knowledge Reasoning over Knowledge Graph" **Expert Systems with Applications** 141: DOI: [10.1016/j.eswa.2019.112948](https://doi.org/10.1016/j.eswa.2019.112948).
- [10] Q. Liu, Q. Zhang, F. Zhao, and G. Wang, (2024) "Uncertain Knowledge Graph Embedding: An Effective Method Combining Multi-Relation and Multi-Path" **Frontiers of Computer Science** 18: DOI: [10.1007/s11704-023-2427-z](https://doi.org/10.1007/s11704-023-2427-z).
- [11] K. Shi, X. Cai, L. Yang, and J. Zhao, (2023) "Enriched Entity Representation of Knowledge Graph for Text Generation" **Complex & Intelligent Systems** 9: 2019–2030. DOI: [10.1007/s40747-022-00898-0](https://doi.org/10.1007/s40747-022-00898-0).
- [12] T. Guo, Q. Yang, C. Wang, Y. Liu, P. Li, J. Tang, D. Li, and Y. Wen, (2024) "KnowledgeNavigator: Leveraging Large Language Models for Enhanced Reasoning over Knowledge Graph" **Complex & Intelligent Systems** 10(5): 7063–7076. DOI: [10.1007/s40747-024-01527-8](https://doi.org/10.1007/s40747-024-01527-8).
- [13] A. Majeed and S. O. Hwang, (2024) "A Multifaceted Survey on Federated Learning: Fundamentals, Paradigm Shifts, Practical Issues, Recent Developments, Partnerships, Trade-Offs, Trustworthiness, and Ways Forward" **IEEE Access** 12: 84643–84679. DOI: [10.1109/ACCESS.2024.3413069](https://doi.org/10.1109/ACCESS.2024.3413069).
- [14] X. Yin, Y. Zhu, and J. Hu, (2021) "A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions" **ACM Computing Surveys** 54(6): DOI: [10.1145/3460427](https://doi.org/10.1145/3460427).
- [15] G. Beigi and H. Liu, (2020) "A Survey on Privacy in Social Media: Identification, Mitigation, and Applications" **ACM Transactions on Data Science** 1(1): DOI: [10.1145/3343038](https://doi.org/10.1145/3343038).