Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Evidence Constrained Agentic Retrieval Augmented Generation for Substation Civil Engineering Preliminary Design Documents in a Single Project Study

Yizhang Huang

huangyizhang2026@163.com

State Grid Shanghai Electric Power Design Co., Ltd.

Xinhui Zhang

State Grid Shanghai Electric Power Design Co., Ltd.

Research Article

Keywords: large language model, retrieval-augmented generation, agentic RAG, substation civil engineering, preliminary design document, regulatory applicability, evidence grounding, cross-section consistency

Posted Date: August 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10423755/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Evidence Constrained Agentic Retrieval Augmented Generation for Substation Civil Engineering Preliminary Design Documents in a Single Project Study

Yizhang Huang[1*] and Xinhui Zhang[1]

[1*]State Grid Shanghai Electric Power Design Co., Ltd., Shanghai, 200000, China .

*Corresponding author(s). E-mail(s): huangyizhang2026@163.com;

Abstract

Substation civil engineering preliminary design documents must integrate project-specific conditions, current regulatory requirements, typical design practices, and historical engineering experience while maintaining consistency across architectural, structural, general-layout, foundation, road, drainage, and technical-indicator sections. Conventional Retrieval-Augmented Generation (RAG) systems remain vulnerable to condition-insensitive retrieval, unequal source authority, incompatible evidence, unsupported engineering claims, and parameter drift during long-document drafting. This study proposes an Evidence-Constrained Agentic Retrieval-Augmented Generation framework, termed EC-ARAG. The framework combines regulation-aware evidence representation, validity-gated hybrid retrieval, source-role-aware conflict resolution, dual planning–retrieval and generation–verification–repair loops, and global project-state memory. The experimental corpus comprised 36 national and industry standards, 12 enterprise standards, 18 typical-design documents, 64 historical preliminary design documents, and nine project-input documents. The held-out evaluation included 120 retrieval queries, 80 applicability cases, 48 evidence-conflict cases, 318 atomic design propositions, and five independent generation runs. EC-ARAG achieved a Recall@5 of 94.1%, an applicability Macro-F1 of 90.6%, an evidence support rate of 95.0%, section completeness of 96.0%, and cross-section parameter consistency of 97.0%. Its unsafe-claim rate and character-level manual-edit rate were 3.4% and 7.6%, respectively. Although agentic verification increased machine inference time, total operational completion time was 21.4% lower than that of closed-corpus Corrective RAG. Under the evaluated single-project, text-only conditions,

EC-ARAG improved the reliability, traceability, and professional editability of regulation-intensive preliminary design drafting.

Keywords: large language model, retrieval-augmented generation, agentic RAG, substation civil engineering, preliminary design document, regulatory applicability, evidence grounding, cross-section consistency

1 Introduction

1.1 Engineering background

Civil engineering preliminary design documents are central deliverables in substation engineering. They provide a textual basis for project review, investment estimation, multidisciplinary coordination, subsequent detailed design, and construction preparation. Their principal content typically includes the project overview and design basis, general layout, functional zoning, architectural scheme, structural system, building and equipment foundations, station roads, vertical planning, drainage, boundary walls, entrances, fire-related civil provisions, and major technical and economic indicators.

Preparing these documents differs from general technical writing in three important respects. First, design statements are constrained by national standards, industry standards, enterprise requirements, typical designs, and project-specific conditions. Second, the applicability of a requirement may depend on the engineering object, building function, voltage level, site condition, structural form, seismic requirement, geotechnical information, design stage, or document version. Third, the same project entities and parameters recur across multiple sections. A building name, floor count, structural system, foundation type, area, road width, or technical indicator must remain consistent throughout the document.

In conventional practice, designers compile civil engineering preliminary design documents by consulting project inputs, standards, typical-design materials, and historical reports. Standardized templates reduce repetitive writing, but substantial manual effort remains necessary to identify applicable clauses, adapt historical descriptions, distinguish mandatory constraints from recommendations, resolve incompatible precedents, verify design parameters, and maintain consistency across sections.

1.2 Opportunities and limitations of large language models

Large Language Models (LLMs) can produce structured technical prose and follow complex drafting instructions. Their parametric knowledge, however, cannot guarantee that an engineering statement is current, applicable, complete, or traceable. Retrieval-Augmented Generation (RAG) connects an LLM with external evidence and conditions generation on retrieved passages [1, 2].

Agent-based methods extend conventional RAG through task decomposition, tool use, adaptive retrieval, reflection, and corrective actions. ReAct coordinates reasoning with external actions [3]; Reflexion uses linguistic feedback and episodic memory [4]; Active RAG determines when and what to retrieve [5]; Self-Refine performs

iterative revision [6]; Self-RAG integrates retrieval and reflection into generation [7]; and Corrective RAG evaluates the quality of initially retrieved information before initiating corrective retrieval [8].

These developments provide a technical foundation for engineering-document drafting, but their direct application to substation preliminary design remains insufficient. A general retriever primarily optimizes semantic relevance, whereas engineering validity requires both relevance and conditional applicability. A clause may address the correct topic but remain invalid because it applies to another building category, site condition, structural form, standard version, or engineering stage.

Knowledge sources also serve different functions. Project inputs define confirmed facts. Current regulations impose mandatory, recommended, permitted, or prohibited requirements. Typical designs provide reusable configurations. Historical documents supply professional expressions and engineering experience but should not override confirmed project facts or current regulations. Treating these sources as interchangeable passages may produce fluent but institutionally invalid conclusions.

Long-document drafting introduces an additional challenge. Conventional RAG generally processes one query at a time and does not necessarily preserve a project parameter after generation moves from general layout to architecture, structure, foundations, and technical indicators. Increasing the context length alone does not guarantee reliable evidence use, particularly when critical information is distributed across many passages [9].

1.3 Research gaps

Three methodological gaps motivate this study. First, existing engineering RAG systems primarily improve document segmentation, hybrid retrieval, graph organization, or question answering. They do not consistently preserve and evaluate the conditions under which a regulatory requirement applies. Semantic similarity therefore cannot reliably distinguish an admissible clause from a conditionally incompatible one.

Second, most engineering RAG studies focus on isolated questions, local compliance decisions, or short technical reports. Preliminary design drafting is a document-level synthesis task. Each section depends on confirmed project facts, regulatory evidence, and information established in previous sections. A section may be linguistically complete while containing unsupported claims or conflicting with another section.

Third, multi-source retrieval is often implemented as passage aggregation. This approach does not explicitly model source authority, version validity, evidence role, or conflict. Historical expressions, typical-design recommendations, enterprise requirements, and mandatory standards may consequently be combined without an engineering priority rule.

1.4 Research questions and contributions

This study addresses four research questions:

RQ1: Can regulation-aware evidence representation and validity-gated retrieval improve evidence recall and clause-applicability classification?

RQ2: Can dual planning–retrieval and generation–verification–repair loops improve engineering-claim support and section completeness?

RQ3: Can source-role-aware conflict resolution and global project-state memory reduce incompatible evidence use and cross-section parameter drift?

RQ4: What trade-off does EC-ARAG introduce between machine inference cost, professional editing effort, and total operational completion time?

The study makes four principal contributions. First, it introduces applicability-aware evidence representation and validity gating. Second, it develops dual-loop agentic RAG with claim-level verification and local repair. Third, it combines source-role-aware conflict resolution with document-level project-state memory. Fourth, it establishes a multi-level evaluation protocol covering evidence extraction, retrieval, applicability, conflict resolution, document quality, human assessment, and computational cost.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 formulates the drafting problem. Section 4 presents EC-ARAG. Section 5 describes the experimental protocol. Section 6 reports the results. Section 7 discusses the findings, engineering implications, and limitations. Section 8 concludes the study.

2 Related Work

2.1 Retrieval-augmented generation and language agents

RAG combines parametric generation with non-parametric external memory [1]. A conventional RAG pipeline segments source documents, encodes passages, retrieves the most relevant candidates, and supplies those candidates to a language model. Subsequent surveys have categorized RAG improvements into retrieval, augmentation, generation, and evaluation components [2].

Agentic approaches introduce adaptive control into this pipeline. ReAct interleaves reasoning with external actions [3]. Reflexion stores verbal feedback in episodic memory [4]. Active RAG initiates retrieval according to the evolving generation context [5], whereas Self-Refine iteratively revises outputs using model-generated feedback [6]. Self-RAG learns when to retrieve and how to critique generated statements [7]. Corrective RAG evaluates initial retrieval quality before corrective acquisition [8]. GraphRAG organizes entities and relationships to support corpus-level information synthesis [10].

These methods motivate iterative retrieval and verification, but their general formulations do not explicitly encode engineering source authority, regulatory applicability, or cross-section document constraints.

2.2 Evaluation of grounded generation

Grounded generation cannot be evaluated solely through lexical similarity. RAGAS evaluates context relevance, faithfulness, and answer relevance [11]. ARES uses lightweight evaluators and limited human annotation to assess context relevance, answer faithfulness, and answer relevance [12]. Systematic RAG evaluations further show that chunking, query transformation, reranking, evidence arrangement, and retrieval depth substantially affect performance [13].

For preliminary design documents, answer-level faithfulness is insufficient. Evaluation must additionally determine whether the retrieved evidence applies to the project, whether the information source has the appropriate engineering role, whether conflicting requirements are safely resolved, and whether repeated project parameters remain consistent across sections.

2.3 RAG in construction and engineering knowledge management

RAG has recently been applied to construction knowledge management, regulatory inquiry, safety information, and engineering design. Wang et al. combined vector representations, a property graph, and keyword retrieval for building-engineering knowledge management [14]. Wu et al. developed hierarchical knowledge organization and retrieval for construction-management question answering [15]. Lee et al. compared RAG and fine-tuned LLMs for construction-safety knowledge retrieval [16]. Uhm et al. evaluated RAG-generated construction-safety information [17], while Baek et al. used a large accident-case corpus and domain-adapted retrieval to generate safety-risk guidance [18].

Engineering-design studies have also explored structured facts and regulatory processing. Siddharth and Luo extracted explicit entity–relation facts from engineering descriptions to ground technical synthesis [19]. Yang and Zhang transformed building-code information into machine-processable representations [20]. Xue et al. developed a question-answering framework for building codes [21]. Shi et al. integrated fine-tuned language modelling and RAG for building-regulation compliance [22]. Xiong et al. combined domain rules, knowledge graphs, and RAG for aviation digital-model design [23].

These studies demonstrate the importance of structured regulatory knowledge. However, most address local questions, compliance decisions, or design assistance rather than coherent multi-section preliminary design drafting.

2.4 Multi-agent engineering and power-engineering systems

Multi-agent LLM systems have been used to coordinate specialized engineering subtasks. Chen and Bao developed a multi-agent framework for code-compliant reinforced-concrete design [24]. Hierarchical multi-agent retrieval has also been investigated for heterogeneous industrial documents [25].

Knowledge graphs and generative models have begun to support electric-power engineering reports, equipment knowledge, and design-parameter recommendation. Qian et al. proposed a knowledge-graph-enhanced framework for power-engineering report generation [26]. Zhou et al. combined a substation knowledge graph with graph neural networks for parameter recommendation [27]. Tang et al. constructed a power-grid equipment knowledge graph to support start-up planning and risk identification [28].

These systems support task specialization and structured knowledge, but they do not jointly address clause applicability, unequal evidence roles, and document-wide consistency in civil preliminary design drafting.

2.5 Methodological distinction

Table 1 compares representative methods with EC-ARAG. “Explicit applicability modelling” indicates that project- or object-specific conditions are represented beyond semantic similarity. “Document-level state” indicates that verified facts are persistently maintained and enforced across multiple generated sections.

As shown in Table 1, the contribution of EC-ARAG is not the isolated use of hybrid retrieval, a knowledge graph, or multiple agents. Its distinction lies in integrating applicability-aware evidence control, source-role-aware conflict resolution, abstention, claim-level verification, and cross-section state enforcement.

Table 1 Comparison with representative related methods.

Study	Primary task	Hybrid/graph	Applicability	Source roles	Conflict	Doc. state	Claim repair
Wang et al. (2025)	Building-engineering QA and generation	Explicit	Not primary	Limited	Not reported	Not reported	Not reported
Wu et al. (2025)	Construction-management QA	Explicit	Not primary	Not reported	Not reported	Not reported	Not reported
Siddharth and Luo (2024)	Engineering knowledge synthesis	Fact based	Not primary	Not reported	Not reported	Not reported	Not reported
Yang and Zhang (2024)	Code transformation	No retrieval focus	Partial	Not reported	Rule level	Not reported	Not reported
Shi et al. (2025)	Code-compliance assistance	Explicit	Partial	Limited	Partial	Not reported	Local
Xiong et al. (2025)	Aviation design assistance	Rule/graph	Partial	Limited	Rule based	Local	Partial
Chen and Bao (2025)	Reinforced-concrete design	Multi-agent	Code oriented	Limited	Partial	Task level	Explicit
Qian et al. (2026)	Power-engineering report generation	Knowledge graph	Not primary	Limited	Not reported	Limited	Not reported
EC-ARAG	Multi-section preliminary design drafting	Explicit	**Explicit**	**Explicit**	**Explicit**	**Explicit**	**Explicit**

3 Problem Definition

3.1 Document scope

The target output is a text-based civil engineering preliminary design document for an anonymized substation project, hereafter referred to as Project X. The evaluated scope includes the project overview and design basis, general layout and functional zoning, architectural design, structural design, building and equipment foundations, station roads and vertical planning, drainage and related civil provisions, boundary walls and entrances, and major technical and economic indicators. Drawings, photographs, point clouds, survey maps, and Building Information Modelling (BIM) models are excluded. Geometric information is represented only through textual project inputs.

3.2 Multi-source knowledge

The knowledge collection is defined as

$$\mathcal{K} = \mathcal{K}_r \cup \mathcal{K}_p \cup \mathcal{K}_t \cup \mathcal{K}_h, \tag{1}$$

where $\mathcal{K}_r$ denotes regulatory knowledge, $\mathcal{K}_p$ denotes project facts, $\mathcal{K}_t$ denotes typical-design knowledge, and $\mathcal{K}_h$ denotes historical-document knowledge. Equation 1 separates the sources because they perform different institutional functions.

3.3 Sections, propositions, and claims

The target document is represented as

$$\mathcal{D} = \{d_1, d_2, \ldots, d_M\}, \tag{2}$$

where d_m is the mth document section. Each section is decomposed into atomic design propositions:

$$d_m = \{z_{m1}, z_{m2}, \ldots, z_{mN_m}\}. \tag{3}$$

A generated proposition may contain several independently verifiable engineering claims:

$$z_{mn} \rightarrow \mathcal{Y}_{mn} = \{y_{mn1}, y_{mn2}, \ldots, y_{mnL_{mn}}\}. \tag{4}$$

Equations 2, 3, and 4 define the hierarchy used for section planning, claim-level verification, and clustered statistical analysis.

3.4 Generation objective

The document-generation task is formulated as

$$Y = G(\mathcal{P}, \mathcal{D}, \mathcal{K}, \Theta), \tag{5}$$

where Y is the generated document, $\mathcal{P}$ is the confirmed project-condition set, and Θ contains retrieval, generation, and agent-control parameters. Equation 5 defines the

overall evidence-conditioned drafting function. The generated document should satisfy

$$\begin{aligned} C_{\mathrm{sup}}(Y) &\geq \tau_{\mathrm{sup}}, \\ C_{\mathrm{app}}(Y) &\geq \tau_{\mathrm{app}}, \\ C_{\mathrm{cov}}(Y) &\geq \tau_{\mathrm{cov}}, \\ C_{\mathrm{state}}(Y) &\geq \tau_{\mathrm{state}}. \end{aligned} \tag{6}$$

Equation 6 expresses the four central requirements: evidence support, regulatory applicability, mandatory-element coverage, and cross-section state consistency.

4 Methodology

4.1 Overall architecture

EC-ARAG consists of six coordinated layers: corpus and project-input processing, regulation-aware evidence construction, validity-gated hybrid retrieval, source-role-aware conflict resolution, dual-loop agent orchestration, and global verification with state updating and document assembly. Figure 1 presents the overall workflow.

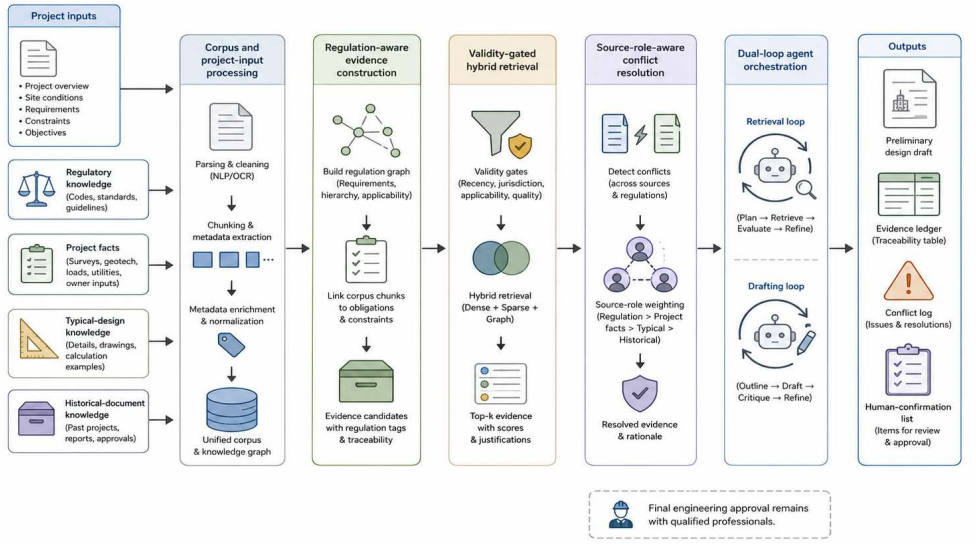


Fig. 1 System boundary and overall EC-ARAG workflow. Project inputs and four textual knowledge sources are processed into structured evidence units. Validity-gated retrieval, conflict resolution, dual-loop generation and verification, and global project-state memory produce a preliminary design draft, evidence ledger, conflict log, and human-confirmation list. Final engineering approval remains the responsibility of qualified professionals.

As shown in Figure 1, the planning agent decomposes each section into atomic propositions. The retrieval router determines the evidence roles required by each proposition. Candidate evidence is obtained through dense, lexical, and graph retrieval, filtered for validity, and examined for conflicts. The generation agent drafts the proposition, after which verification and repair operate at claim level. Verified facts are subsequently admitted to global project-state memory.

4.2 Corpus processing and project isolation

Regulatory documents were segmented according to complete clause boundaries. Definitions, notes, exceptions, referenced clauses, and tabular conditions were linked to their principal requirements. Fixed-length segmentation was avoided when it would separate a numerical value from its condition or exception. Typical-design documents were segmented by engineering object and design recommendation, whereas historical preliminary design documents were segmented by section and semantic paragraph.

The initial historical archive contained 92 files. Eight exact duplicates were removed using file hashes. Seven files associated with Project X, five revised or derivative files, and eight near-duplicate files were also excluded. The final historical corpus therefore contained 64 documents. Near-duplicate screening used BGE-M3 document embeddings. Nineteen candidate document pairs exceeded the predefined cosine-similarity threshold of 0.92 and were manually reviewed; eight documents were subsequently removed. The completed Project X reference document was excluded from the retrieval corpus, section-schema construction, prompt development, query formulation, parameter selection, and agent-template design. Figure 2 summarizes the process.

4.3 Evidence-unit construction and quality control

A regulation evidence unit is represented as

$$e_i = (t_i, o_i, \mathcal{C}_i, a_i, v_i, u_i, x_i, \lambda_i, s_i, \nu_i, d_i, \mathcal{R}_i^{\mathrm{ref}}), \tag{7}$$

where t_i is the original clause text, o_i is the constrained engineering object, $\mathcal{C}_i$ is the applicability-condition set, a_i is the prescribed action, v_i is a numerical value or interval, u_i is the unit, x_i contains exceptions, λ_i denotes obligation modality, s_i is the source and authority class, ν_i is the version and validity state, d_i is the mapped section set, and $\mathcal{R}_i^{\mathrm{ref}}$ contains related clause references. Equation 7 prevents values, conditions, and exceptions from being separated during retrieval.

A project fact is represented as

$$p_j = (\epsilon_j, a_j^{\mathrm{attr}}, \omega_j, u_j, \sigma_j, d_j, \kappa_j), \tag{8}$$

where ϵ_j is the project entity, a_j^{attr} is the attribute, ω_j is its value, σ_j is the provenance, and κ_j is the confirmation state. Equation 8 supports confirmed, derived, provisional, and rejected states.

A section requirement is defined as

$$q_m = (g_m, b_m, h_m, l_m, e_m), \tag{9}$$

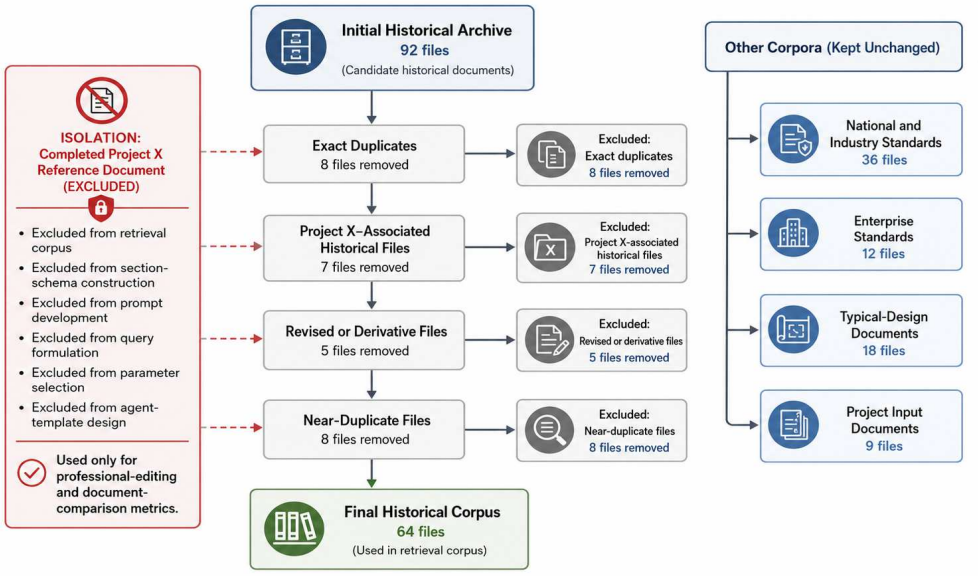


Fig. 2 Corpus selection and project-level leakage control. The initial archive of 92 historical documents was reduced to 64 after removing exact duplicates, Project X-associated files, revised or derivative versions, and near-duplicate documents. The completed Project X reference document was isolated from retrieval, development, prompt design, and parameter selection.

where g_m is the section objective, b_m is the mandatory-element set, h_m is the prerequisite-fact set, l_m is the inter-section dependency set, and e_m specifies the required evidence roles. Equation 9 formalizes the coverage and dependency constraints applied during section planning.

Table 2 summarizes the unit types, and Figure 3 shows the property-graph organization.

A schema-constrained extraction prompt generated candidate evidence fields. Deterministic parsers handled clause identifiers, numerical values, units, and version metadata. Two engineering annotators reviewed the extracted units through a structured interface. Units containing unresolved cross-references or ambiguous exceptions were subjected to additional review.

4.4 Applicability matching and validity gating

For condition k of evidence unit e_i, the project-condition match is

$$m_{ik} = \begin{cases} 1, & \text{confirmed match,} \\ 0.5, & \text{condition unknown,} \\ 0, & \text{confirmed contradiction.} \end{cases} \tag{10}$$

Table 2 Knowledge-unit schema.

Unit type	Principal fields	Primary function
Regulation evidence	Clause, object, condition, action, value, unit, exception, modality, authority, version	Defines valid engineering constraints
Project fact	Entity, attribute, value, unit, provenance, section, state	Defines Project X conditions
Section requirement	Objective, mandatory elements, prerequisites, dependencies, evidence roles	Controls coverage and generation order
Typical-design evidence	Configuration, applicable project class, recommendation level, source	Supplies reusable design patterns
Historical evidence	Project context, section type, terminology, expression pattern	Supplies professional expression

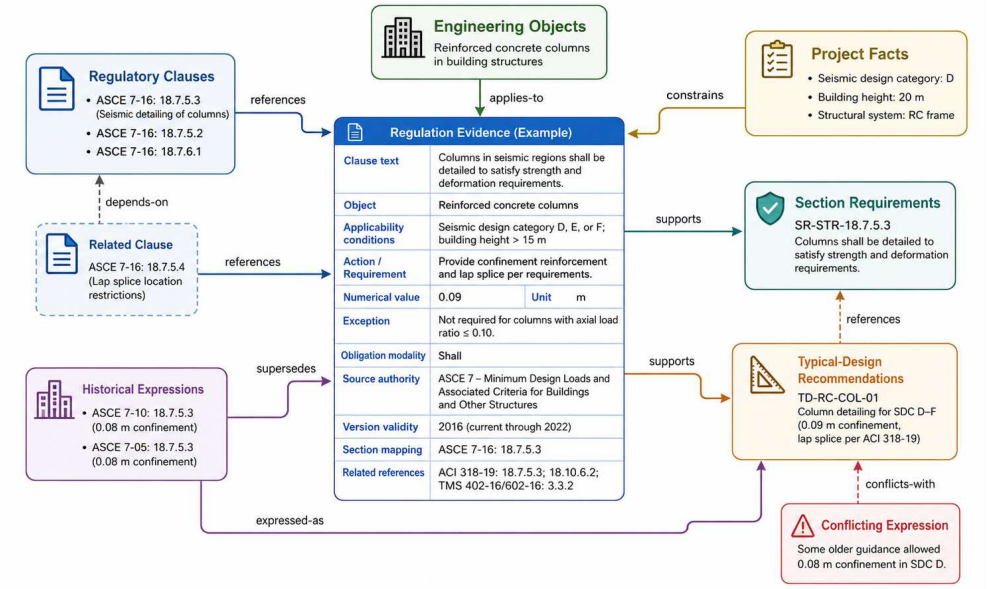


Fig. 3 Regulation-aware evidence graph and example evidence unit. The graph links regulatory clauses, engineering objects, project facts, section requirements, typical-design recommendations, and historical expressions. The evidence card preserves the original clause, conditions, requirement, value, unit, exception, authority, version, and target section.

The soft applicability score is

$$S_{\text{app}}(e_i, \mathcal{P}) = \frac{\sum_{k=1}^{K_i} w_{ik} m_{ik}}{\sum_{k=1}^{K_i} w_{ik}}, \tag{11}$$

where w_{ik} denotes the importance of condition k. Equations 10 and 11 distinguish confirmed matches from unresolved conditions.

Validity is determined before relevance ranking:

$$\mathcal{E}_{\text{valid}}(q, \mathcal{P}) = \left\{ e_i \in \mathcal{E} : V_i = 1,\ D_i \sim D_q,\ \nexists c_{ik}^{H} \in \mathcal{C}_i \text{ with } m_{ik} = 0 \right\}. \tag{12}$$

In Equation 12, $V_i = 1$ indicates a valid source version, $D_i \sim D_q$ indicates discipline compatibility, and c_{ik}^{H} denotes a hard applicability condition. An obsolete clause or contradicted hard condition is therefore excluded rather than merely down-weighted.

4.5 Source-aware hybrid retrieval

Candidate evidence is retrieved through dense semantic search, BM25 lexical search, and graph-neighbour expansion. The BM25 score is [29]

$$S_{\text{BM25}}(q, e) = \sum_{w \in q} \text{IDF}(w) \frac{f(w, e)(k_1 + 1)}{f(w, e) + k_1 \left(1 - b + b \frac{|e|}{\text{avgdl}}\right)}, \tag{13}$$

where $f(w, e)$ is term frequency, $|e|$ is evidence length, and avgdl is average evidence length. Equation 13 preserves exact terminology, identifiers, values, and units during sparse retrieval.

After validity gating, candidates are ranked using

$$\begin{aligned} S_{\text{rank}}(e_i, q, \mathcal{P}) = \alpha S_{\text{dense}} + \beta S_{\text{BM25}} + \gamma S_{\text{graph}} \\ + \delta S_{\text{section}} + \eta S_{\text{authority}} + \zeta S_{\text{app}}, \end{aligned} \tag{14}$$

subject to $\alpha + \beta + \gamma + \delta + \eta + \zeta = 1$. The adopted values were $\alpha = 0.30$, $\beta = 0.20$, $\gamma = 0.12$, $\delta = 0.10$, $\eta = 0.10$, and $\zeta = 0.18$. Equation 14 combines semantic, lexical, graph, section, authority, and applicability signals.

BGE-M3 supplied normalized dense embeddings [30]. The retrieval design follows established sentence- and dense-representation principles [31, 32], while the final reranking stage uses a cross-encoder architecture consistent with BERT passage reranking [33].

4.6 Evidence-conflict detection and resolution

Two evidence units are considered conflicting when they constrain the same object and attribute under overlapping conditions but impose incompatible normalized requirements:

$$\chi(e_i, e_j) = \mathbb{I}(o_i = o_j)\mathbb{I}(a_i^{\text{attr}} = a_j^{\text{attr}})\mathbb{I}(\mathcal{C}_i \cap \mathcal{C}_j \neq \varnothing)\, \text{Contradict}(\tilde{r}_i, \tilde{r}_j). \tag{15}$$

Equation 15 handles numerical, categorical, modal, version, and source-role conflicts.

The priority score for a valid conflict candidate is

$$P(e_i) = \rho_1 A_i + \rho_2 R_i^{\text{recency}} + \rho_3 S_{\text{app}}(e_i, \mathcal{P}) + \rho_4 G_i + \rho_5 S_{\text{rank}}(e_i, q, \mathcal{P}), \tag{16}$$

where A_i is authority, R_i^{recency} is relative priority among valid versions, and G_i is requirement specificity. Equation 16 produces selection of a dominant evidence unit, retention of a compatible subset, or abstention. A priority-margin value below 0.08 triggers abstention.

4.7 Dual-loop agent architecture

Figure 4 presents the two interacting loops. The outer loop manages planning and evidence acquisition, whereas the inner loop manages generation, verification, and local repair.

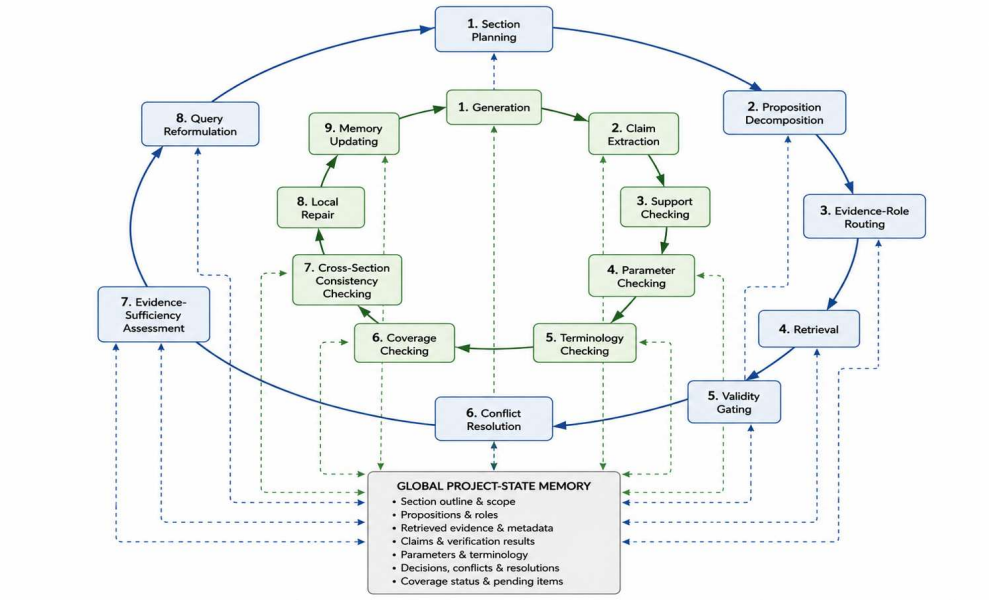


Fig. 4 Dual planning–retrieval and generation–verification–repair loops. The outer loop performs section planning, proposition decomposition, evidence-role routing, retrieval, validity gating, conflict resolution, evidence-sufficiency assessment, and query reformulation. The inner loop performs generation, claim extraction, support checking, parameter checking, terminology checking, coverage checking, cross-section checking, local repair, and memory updating.

For proposition z, let $\mathcal{R}_z$ denote the required evidence-role set. Evidence sufficiency is

$$E_{\text{suf}}(z) = \frac{1}{|\mathcal{R}_z|} \sum_{r \in \mathcal{R}_z} \max_{\substack{e \in \mathcal{E}_z \\ \text{role}(e)=r}} \left[S_{\text{rank}}(e, z) \mathbb{I}(e \models z) \right]. \tag{17}$$

The score in Equation 17 lies in $[0, 1]$. Retrieval is repeated when $E_{\text{suf}}(z) < 0.78$, with a maximum of three iterations.

For generated claim y, the composite verification-deficiency score is

$$D_{\text{ver}}(y) = \mu_1 D_{\text{evi}} + \mu_2 D_{\text{par}} + \mu_3 D_{\text{ter}} + \mu_4 D_{\text{cov}} + \mu_5 D_{\text{cross}}, \tag{18}$$

where the five components represent evidence, parameter, terminology, coverage, and cross-section deficiencies. Equation 18 combines these complementary failure signals into the repair trigger. The adopted weights were 0.35, 0.20, 0.10, 0.15, and 0.20, respectively. A claim is repaired when $D_{\text{ver}} > 0.20$ or when a hard violation is detected.

Table 3 specifies the functional agents. Algorithm 1 presents the complete workflow.

Algorithm 1 Evidence-constrained preliminary design drafting

Require: Project facts $\mathcal{P}$, document structure $\mathcal{D}$, knowledge collection $\mathcal{K}$, sufficiency threshold τ_{suf}, verification threshold τ_{ver}

Ensure: Draft Y, evidence ledger, conflict log, and human-confirmation list

```
Initialize global project-state memory from confirmed inputs
Construct the section-dependency order
for all sections d_m ∈ D do
    Generate atomic design propositions
    for all propositions z do
        Determine required evidence roles
        Retrieve dense, lexical, and graph candidates
        Apply validity and hard-condition filters
        Rerank valid evidence using Equation 14
        Detect and resolve conflicts using Equation 15
        Calculate evidence sufficiency using Equation 17
        if E_suf(z) < τ_suf then
            Reformulate the query and repeat retrieval, subject to the iteration limit
        end if
        Generate a proposition from admissible evidence and locked facts
        Divide the proposition into independently verifiable claims
        Verify support, applicability, parameters, terminology, coverage, and consistency
        if D_ver > τ_ver or a hard violation is detected then
            Repair only the failed span, subject to the repair limit
        end if
        Admit confirmed and fully verified derived facts into memory
    end for
end for
Assemble verified sections and perform final document-level checking
Export the draft, evidence ledger, unresolved-condition list, and revision log
```

4.8 Global project-state memory and stopping criterion

At generation step t, the memory is represented as

$$\mathcal{M}^{(t)} = \{(\epsilon, a^{\text{attr}}, \omega, u, \kappa, \pi, \mathcal{S}, \ell)\}, \tag{19}$$

Table 3 Agent specifications.

Agent	Input	Output	Decision mechanism	Failure action
Planning agent	Section schema and project state	Atomic propositions	LLM with JSON schema	Re-decompose section
Retrieval router	Proposition and evidence-role requirements	Retrieval plan	Rules and LLM classifier	Activate all relevant routes
Regulation retriever	Query and project conditions	Valid regulation evidence	Dense, BM25, graph, and filters	Reformulate query
Project retriever	Entity and attribute	Confirmed project facts	Structured database search	Mark condition as missing
Conflict-resolution agent	Competing evidence	Selected evidence or abstention	Priority rules and LLM explanation	Create confirmation item
Generation agent	Evidence bundle and locked state	Draft paragraph	Qwen2.5-32B-Instruct	Transfer to verification
Verification agent	Claims, evidence, and project state	Error type and location	Independent verifier and rules	Trigger local repair
Repair agent	Failed span, error, and evidence	Revised span	Qwen2.5-32B-Instruct	Abstain after limit
State manager	Verified candidate facts	State-update decision	Deterministic write gate	Reject conflicting write

where ϵ is an entity, a^{attr} is an attribute, ω is the value, κ is the confirmation state, π is provenance, $\mathcal{S}$ is the set of sections using the fact, and ℓ is the lock state.

A candidate fact $\hat{p}$ is processed according to

$$U(\hat{p}, \mathcal{M}) = \begin{cases} \text{lock}, & \kappa = \text{confirmed}, \\ \text{admit}, & \kappa = \text{derived and fully verified}, \\ \text{hold}, & \kappa = \text{provisional}, \\ \text{reject}, & \hat{p} \text{ conflicts with a locked item}. \end{cases} \tag{20}$$

Equations 19 and 20 preserve stable project entities and prevent unverified values from propagating to later sections.

The aggregate document deficiency after iteration t is

$$D^{(t)} = \omega_1(1 - C_{\text{sup}}) + \omega_2(1 - C_{\text{app}}) + \omega_3(1 - C_{\text{cov}}) + \omega_4(1 - C_{\text{state}}). \tag{21}$$

Generation terminates when

$$D^{(t)} \leq \varepsilon \quad \text{or} \quad t = T_{\max}. \tag{22}$$

Equations 21 and 22 ensure that unresolved conditions remain visible rather than being silently converted into design conclusions.

4.9 Implementation environment

Qwen2.5-32B-Instruct was used as the generator and principal reasoning model [34]. BGE-M3 was used for dense representations, and BGE-reranker-v2-m3 was used for cross-encoder reranking [30]. The independent semantic verifier used Qwen2.5-7B-Instruct with deterministic decoding.

Experiments were conducted on a Linux workstation equipped with two NVIDIA A100 80-GB GPUs, 256 GB of system memory, and an AMD EPYC processor. Table 4 reports the final configuration.

All settings in Table 4 were frozen before the held-out test set was evaluated.

Table 4 Implementation configuration.

Component	Configuration
Generator and principal agent model	Qwen2.5-32B-Instruct
Independent semantic verifier	Qwen2.5-7B-Instruct
Embedding model / reranker	BGE-M3 / BGE-reranker-v2-m3
Dense index / graph store	FAISS inner-product index / Neo4j property graph
Sparse retrieval	BM25, $k_1 = 1.5$, $b = 0.75$
Regulation segmentation	Complete clauses with linked notes, definitions, and exceptions
Historical-document chunking	384 tokens with 64-token overlap
Candidate sizes	Dense 40, sparse 40, reranking 30, final evidence 10
Graph expansion / evidence context	Two hops / 8,000 tokens per section
Generation parameters	Temperature 0.20, top-p 0.85, repetition penalty 1.05
Control thresholds	Sufficiency 0.78, conflict margin 0.08, verification 0.20
Iteration limits	Retrieval 3, repair 2
Software	Ubuntu 22.04, Python 3.10.13, PyTorch 2.4.1, CUDA 12.1, Transformers 4.46.3, vLLM 0.6.4, FAISS 1.8.0, Neo4j 5.25
Experimental seeds	21, 32, 42, 56, and 73

5 Experimental Design

5.1 Corpus and project isolation

The study used textual materials only. The corpus contained 36 national and industry standards, 12 enterprise standards, 18 typical-design documents, 64 historical preliminary design documents, and nine Project X input documents. Table 5 presents the corpus statistics and isolation results.

As reported in Table 5, 28 historical files were excluded before indexing. The final retrievable corpus contained 13,634 regulatory, typical-design, and historical evidence units, excluding structured project facts and section requirements.

5.2 Development and held-out test sets

The query-level benchmark was divided before model evaluation. The development set was used exclusively to determine retrieval weights, thresholds, and repair limits. The held-out test set was used only for final reporting. Table 6 reports the partition.

All thresholds and weights reported in Section 4.9 were fixed before the Project X generation experiment.

5.3 Evidence-extraction evaluation

A stratified sample of 320 regulatory evidence units was independently annotated by two civil-engineering reviewers. Disagreements were resolved by a senior reviewer, and agreement was calculated before adjudication. Table 7 reports field-level performance.

Table 7 shows that numerical values, units, and source metadata were extracted reliably, whereas applicability conditions and exceptions were more difficult. Inter-annotator agreement for the complete schema was $\kappa = 0.86$ (95% CI [0.82, 0.89]).

Table 5 Corpus composition and project-isolation results.

Category	Initial	Excluded	Final	Role
National and industry standards	36	0	36	Regulatory constraints
Enterprise standards	12	0	12	Organization-specific requirements
Typical-design documents	18	0	18	Recommended configurations
Historical preliminary design files	92	28	64	Engineering expression and experience
Exact historical duplicates	8	8	0	Removed
Project X-associated historical files	7	7	0	Removed
Revised or derivative historical files	5	5	0	Removed
Near-duplicate historical files	8	8	0	Removed
Project X input documents	9	0	9	Confirmed project conditions
Regulation evidence units	–	–	5,486	Clause-level evidence
Typical-design evidence units	–	–	1,236	Configuration evidence
Historical-expression units	–	–	6,912	Professional-language evidence
Confirmed project facts	–	–	228	Entity–attribute–value records
Section requirement units	–	–	96	Coverage and dependency constraints

Table 6 Development and held-out test partitions.

Task	Total	Development	Held-out test	Primary use
Retrieval queries	180	60	120	Retrieval weighting and final Recall/MRR
Applicability cases	120	40	80	Calibration and final Macro-F1
Evidence-conflict cases	72	24	48	Conflict-margin calibration and evaluation
Evidence-extraction audit	320	–	320	Independent extraction-quality evaluation
Atomic design propositions	318	Prompt pilot only	318	Document-generation evaluation

Table 7 Regulation-evidence field extraction.

Field	Precision (%)	Recall (%)	F1 (%)
Engineering object	94.8	93.6	94.2
Applicability condition	88.9	86.7	87.8
Requirement/action	95.1	94.0	94.5
Numerical value	98.1	97.4	97.7
Unit	98.6	98.0	98.3
Obligation modality	91.7	90.5	91.1
Exception	84.6	81.3	82.9
Source and version	99.1	98.8	98.9
Section mapping	92.3	90.8	91.5

Automatic parsing of the 5,486 regulation units required 2.8 GPU-hours, and complete human review required 31.2 person-hours. A total of 811 units (14.8%) required at least one field correction.

5.4 Baselines and fairness controls

Seven methods were compared: template-based drafting, LLM-only generation, Dense RAG, Hybrid RAG, GraphRAG, Closed-corpus Corrective RAG, and EC-ARAG. The GraphRAG baseline used local search rather than global community summarization because the evaluation required clause- and entity-level evidence. Document GraphRAG was consulted when adapting graph-context construction to document question answering [35]. Closed-corpus Corrective RAG used the same BGE-M3 retriever and reranker as Hybrid RAG and performed one query-reformulation step when the initial evidence-quality score was below 0.72. No web search or additional external corpus was permitted.

Table 8 summarizes the controls.

All retrieval-based methods used the same corpus, generator, maximum section length, and evidence-token limit. As shown in Table 8, EC-ARAG's additional resources consisted of control mechanisms rather than access to additional documents.

5.5 Expert and annotation protocol

The evaluation comprised two separate blinded stages. In the document-quality stage, six reviewers evaluated all seven anonymized documents in independently randomized order. Seed 42 was specified before any output was inspected and was used for every method. Documents were presented with identical typography and without citations, evidence ledgers, method names, or agent logs. The dimensions were engineering applicability, professional expression, section completeness, internal consistency, and practical editability.

In the evidence-auditing stage, claims, source identifiers, excerpts, applicability conditions, and citation locations were presented in a uniform interface. Two trained annotators independently evaluated claim support, applicability, conflict decisions, and unsafe claims. A stratified sample of 240 claims was additionally triple-coded to estimate multi-rater reliability.

The expert-evaluation protocol was approved by the Ethics Committee of State Grid Shanghai Electric Power Design Co., Ltd. and was conducted in accordance with the Declaration of Helsinki and the applicable institutional guidelines and regulations for research involving human participants. All reviewers and annotators received information about the study procedures before participation and provided written informed consent.

5.6 Evaluation metrics

Recall at k is

$$\text{Recall@}\,k = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathcal{E}_i^* \cap \mathcal{E}_i^k|}{|\mathcal{E}_i^*|}, \tag{23}$$

where $\mathcal{E}_i^*$ is the gold evidence set and $\mathcal{E}_i^k$ is the retrieved top-k set. Mean Reciprocal Rank is

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i}, \tag{24}$$

where r_i is the rank of the first relevant evidence unit.

For applicability class c,

$$F1_c = \frac{2P_cR_c}{P_c + R_c}, \tag{25}$$

and three-class applicability Macro-F1 is

$$\text{MacroF1} = \frac{1}{3}\sum_{c=1}^{3}F1_c. \tag{26}$$

Table 8 Baseline design and fairness controls.

Method		Retrieval	Iterative correction	Applicability metadata	Conflict resolution	Persistent state	Token limit
Template-based		None	No	Template rules only	No	Manual insertion	–
LLM-only		None	No	No	No	Prompt context	–
Dense RAG		Dense	No	No	No	No	8,000
Hybrid RAG		Dense + BM25 + reranker	No	No	No	No	8,000
GraphRAG		Dense + local graph context	No	No	Graph adjacency only	No	8,000
Closed-corpus RAG	Corrective	Hybrid retrieval	One corrective pass	No	Retrieval-quality correction	No	8,000
EC-ARAG		Validity-gated hybrid retrieval	Retrieval and repair loops	Yes	Explicit	Explicit	8,000

Equations 23, 24, 25, and 26 quantify retrieval coverage, ranking quality, class-specific performance, and applicability discrimination.

The evidence support rate is

$$R_{\text{sup}} = \frac{N_{\text{supported and applicable}}}{N_{\text{engineering claims}}}, \tag{27}$$

and the unsafe-claim rate is

$$R_{\text{unsafe}} = \frac{N_{\text{unsupported}} + N_{\text{contradicted}} + N_{\text{inapplicable}}}{N_{\text{engineering claims}}}. \tag{28}$$

Equations 27 and 28 distinguish applicable evidence support from unsupported, contradicted, or inapplicable claims. Condition-unverifiable claims were reported separately.

Section completeness is

$$C_{\text{sec}} = \frac{N_{\text{covered mandatory elements}}}{N_{\text{required mandatory elements}}}, \tag{29}$$

Equation 29 measures mandatory-content coverage, and parameter consistency is

$$C_{\text{par}} = \frac{\sum_{g=1}^{G} \mathbb{I}(|\operatorname{Unique}(V_g)| = 1)}{G}. \tag{30}$$

Equation 30 evaluates parameter groups that appear in at least two sections.

Because the outputs were primarily Chinese, the manual-edit rate was calculated at character level:

$$R_{\text{edit}} = \frac{D_{\text{Lev}}^{\text{char}}(Y_{\text{generated}}, Y_{\text{approved}})}{|Y_{\text{approved}}|_{\text{char}}}. \tag{31}$$

Equation 31 measures the character-level revision burden; formatting differences were excluded. Operational completion time is

$$T_{\text{operational}} = T_{\text{generation}} + T_{\text{professional review}}. \tag{32}$$

Equation 32 excludes the one-time corpus-construction cost.

Overall claim-level metrics were micro-averaged:

$$R_{\text{micro}} = \frac{\sum_s N_{\text{supported},s}}{\sum_s N_{\text{claims},s}}. \tag{33}$$

Equation 33 weights sections by their claim counts. Section-level results were reported independently and were not equally weighted to derive the overall value.

5.7 Statistical analysis

Stochastic methods were run using five random seeds. Metrics were calculated separately for each run and are reported as run-level mean ± standard deviation. Query-level confidence intervals were obtained through paired bootstrap resampling with 10,000 iterations. Claim-level confidence intervals used clustered bootstrap resampling at the atomic-proposition level.

A mixed-effects logistic model was used as a robustness analysis:

$$\text{logit}[P(Y_{irsm} = 1)] = \beta_0 + \beta_1 M_m + u_r + v_s + w_i, \tag{34}$$

where M_m is the method effect, u_r is a run-level random intercept, v_s is a section-level random intercept, and w_i is a proposition-level random intercept. Equation 34 accounts for clustering across runs, sections, and propositions. Expert ratings were evaluated using a Friedman test followed by Holm-adjusted Wilcoxon signed-rank tests. Statistical significance was set at $p < 0.05$.

6 Results

6.1 Claim counts across independent runs

Table 9 reports the number and disposition of claims in each EC-ARAG run. The detailed error analysis uses seed 42, which was fixed before output inspection.

Table 9 Claim counts across EC-ARAG runs.

Seed	Total	Supported	Unsafe	Unverifiable	Support (%)
21	1,605	1,519	57	29	94.64
32	1,621	1,541	55	25	95.06
42	1,612	1,531	55	26	94.98
56	1,608	1,525	56	27	94.84
73	1,617	1,539	52	26	95.18

As shown in Table 9, the number of claims varied from 1,605 to 1,621, while the support rate remained between 94.64% and 95.18%.

6.2 Evidence-extraction performance

The extraction results in Table 7 indicate that numerical and metadata fields were more reliable than conditional and exception information. Exception extraction obtained the lowest F1 score, at 82.9%. Among sampled exception errors, 45.2% involved table notes, 29.0% involved references to another clause, and 25.8% involved nested conditional language. Downstream tracing showed that 11 of the 58 residual error instances in the seed-42 output originated from evidence parsing.

6.3 Overall performance

Table 10 reports retrieval, applicability, and document-generation performance. Figure 5 visualizes the retrieval-based methods.

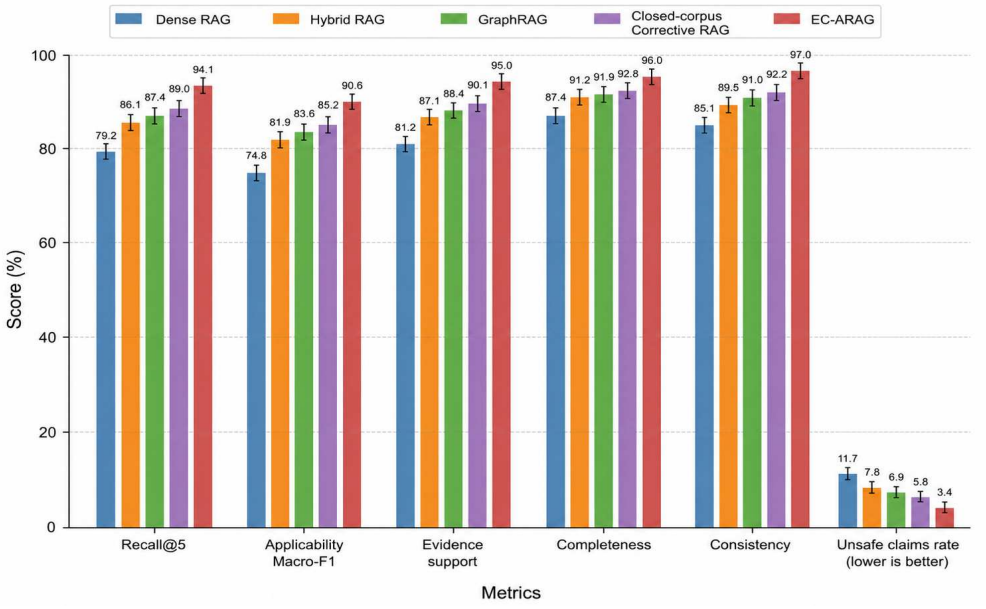


Fig. 5 Overall performance of retrieval-based methods. Grouped bars compare Recall@5, applicability Macro-F1, evidence support, section completeness, parameter consistency, and the lower-is-better unsafe-claim rate.

EC-ARAG achieved a Recall@5 of 94.1%, exceeding Closed-corpus Corrective RAG, GraphRAG, and Hybrid RAG by 5.1, 6.7, and 8.0 percentage points, respectively. Its MRR of 0.892 indicates that relevant evidence was generally positioned near the beginning of the retrieved context. Applicability Macro-F1 reached 90.6%. The 5.4-percentage-point difference from the strongest baseline was slightly larger than the Recall@5 difference, indicating improved discrimination between applicable and conditionally invalid clauses.

EC-ARAG achieved an evidence support rate of 95.0% and an unsafe-claim rate of 3.4%. Condition-unverifiable claims accounted for a further 1.6% and were transferred to the human-confirmation list. Figure 5 shows that the largest gains occurred in applicability, consistency, and unsafe-claim reduction rather than in completeness alone.

6.4 Statistical significance

Table 11 compares EC-ARAG with Closed-corpus Corrective RAG, the strongest baseline.

Table 10 Overall performance.

Method	Recall@5 (%)	MRR	Macro-F1 (%)	Support (%)	Complete (%)	Consistent (%)	Unsafe ↓ (%)	Edit ↓ (%)
Template-based	–	–	–	74.2	84.7	88.5	14.9	24.8
LLM-only	–	–	–	70.1 ± 2.1	81.9 ± 1.9	79.4 ± 2.4	19.3 ± 1.6	27.6 ± 2.3
Dense RAG	79.2 ± 1.3	0.721 ± 0.012	74.8 ± 1.6	81.2 ± 1.5	87.4 ± 1.3	85.1 ± 1.7	11.7 ± 1.0	18.5 ± 1.6
Hybrid RAG	86.1 ± 1.1	0.803 ± 0.010	81.9 ± 1.4	87.1 ± 1.2	91.2 ± 1.1	89.5 ± 1.3	7.8 ± 0.8	14.3 ± 1.2
GraphRAG	87.4 ± 1.0	0.821 ± 0.009	83.6 ± 1.3	88.4 ± 1.1	91.9 ± 1.0	91.0 ± 1.1	6.9 ± 0.7	13.2 ± 1.1
Closed-corpus Corrective RAG	89.0 ± 0.9	0.839 ± 0.008	85.2 ± 1.1	90.1 ± 1.0	92.8 ± 0.9	92.2 ± 1.0	5.8 ± 0.6	11.8 ± 0.9
EC-ARAG	**94.1 ± 0.7**	**0.892 ± 0.006**	**90.6 ± 0.9**	**95.0 ± 0.8**	**96.0 ± 0.7**	**97.0 ± 0.6**	**3.4 ± 0.5**	**7.6 ± 0.8**

Table 11 Statistical comparison with the strongest baseline.

Metric	Absolute difference	95% confidence interval	Adjusted p	Effect size
Recall@5	+5.1 percentage points	[3.7, 6.5]	< 0.001	Cliff's $\delta = 0.71$
Applicability Macro-F1	+5.4 percentage points	[3.9, 6.8]	< 0.001	Cliff's $\delta = 0.74$
Evidence support	+4.9 percentage points	[3.2, 6.6]	< 0.001	Odds ratio = 2.08
Completeness	+3.2 percentage points	[1.8, 4.5]	0.002	Cliff's $\delta = 0.59$
Parameter consistency	+4.8 percentage points	[3.3, 6.2]	< 0.001	Odds ratio = 2.43
Unsafe-claim rate	-2.4 percentage points	[-3.5, -1.4]	< 0.001	Odds ratio = 0.57
Character-level edit rate	-4.2 percentage points	[-5.6, -2.8]	< 0.001	Cliff's $\delta = -0.68$
Overall expert score	+0.31/5	[0.20, 0.42]	0.002	Rank-biserial $r = 0.61$

All differences in Table 11 remained statistically significant after Holm correction. Mixed-effects robustness analysis produced the same direction and significance for evidence support and unsafe claims. The effect size for completeness was smaller than those for applicability and consistency.

6.5 Ablation analysis

Table 12 and Figure 6 isolate the contribution of each principal component.

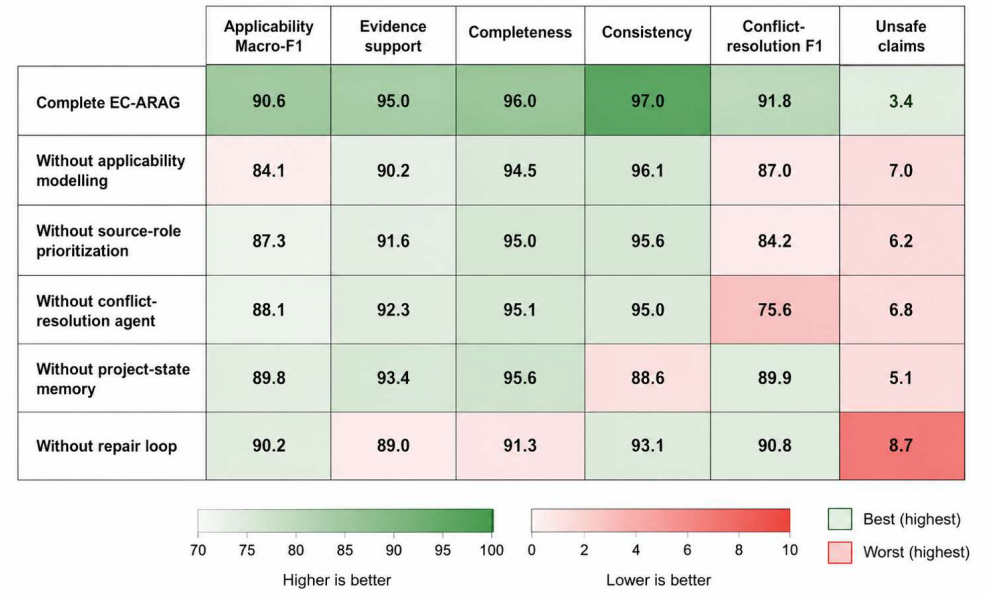


Fig. 6 Ablation effects on major quality dimensions. The heat map compares applicability Macro-F1, evidence support, completeness, consistency, conflict-resolution F1, and unsafe claims for the complete method and five ablated variants.

Removing applicability modelling reduced Macro-F1 by 6.5 percentage points and increased unsafe claims from 3.4% to 7.0%. Removing the conflict-resolution agent caused conflict-resolution F1 to decrease from 91.8% to 75.6%, while completeness remained above 95%. Removing project-state memory reduced consistency from 97.0% to 88.6%. Removing the repair loop increased unsafe claims to 8.7%. Figure 6 demonstrates that the modules address distinct rather than interchangeable failure modes.

6.6 Section-level performance

Table 13 reports seed-42 claim counts and section-level results. Figure 7 presents the same results graphically.

Architectural design achieved the highest evidence support rate. Roads, vertical planning, and drainage achieved the lowest completeness and highest edit rate because

Table 12 Ablation results.

Variant	Applicability F1 (%)	Support (%)	Complete (%)	Consistent (%)	Conflict F1 (%)	Unsafe ↓ (%)
Complete EC-ARAG	90.6	95.0	96.0	97.0	91.8	3.4
Without applicability modelling	84.1	90.2	94.5	96.1	87.0	7.0
Without source-role prioritization	87.3	91.6	95.0	95.6	84.2	6.2
Without conflict-resolution agent	88.1	92.3	95.1	95.0	75.6	6.8
Without project-state memory	89.8	93.4	95.6	88.6	89.9	5.1
Without repair loop	90.2	89.0	91.3	93.1	90.8	8.7

Table 13 Section-level claim outcomes and document quality.

Section	Claims	Supported	Unsafe	Unverifiable	Support (%)	Complete (%)	Consistent (%)	Expert /5	Edit ↓ (%)
General layout	270	254	10	6	94.1	94.2	96.1	4.44	8.9
Architectural design	285	277	5	3	97.2	97.5	97.8	4.61	6.2
Structural design	280	268	8	4	95.7	95.4	96.8	4.53	7.4
Building and equipment foundations	240	225	10	5	93.8	93.6	95.4	4.37	9.8
Roads, vertical planning, and drainage	255	238	12	5	93.3	92.7	94.8	4.31	10.6
Technical and economic indicators	282	269	10	3	95.4	96.8	99.0	4.66	4.2
Total/micro-average	1,612	1,531	55	26	95.0	–	–	–	–

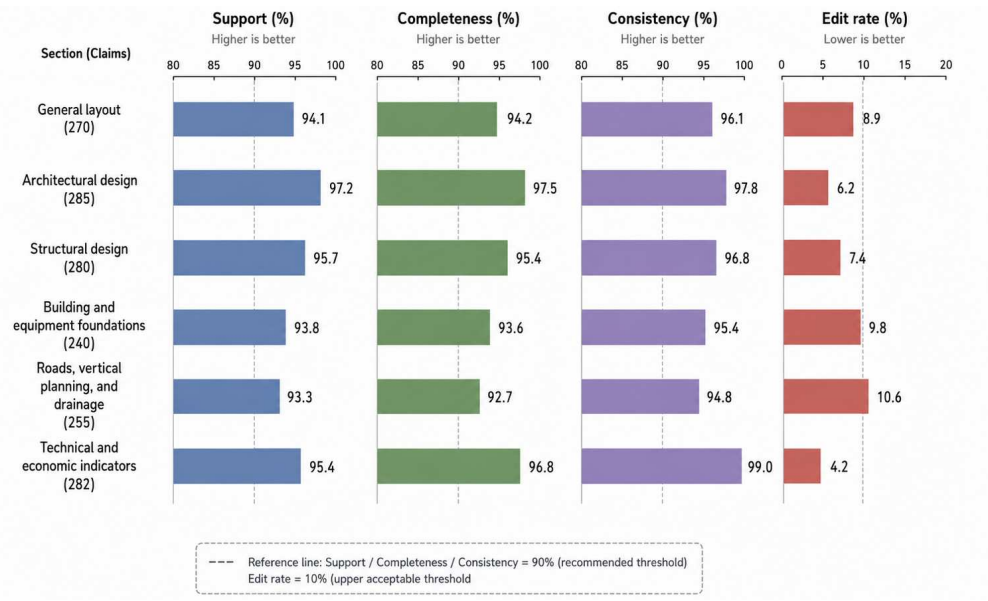


Fig. 7 Section-level quality and editability. Horizontal bars compare evidence support, completeness, consistency, and edit rate across six section groups. Claim counts are reported beside the section names.

these sections depend strongly on geometric relationships, survey elevations, road gradients, drainage directions, and discharge boundaries. Technical and economic indicators achieved 99.0% consistency because repeated values were retrieved from locked project-state records rather than independently regenerated.

6.7 Verification-agent performance

The independent verifier was evaluated using a stratified set of 400 manually annotated claims. Table 14 reports its performance.

Table 14 Verification-component performance.

Verification component	Precision (%)	Recall (%)	F1 (%)
Evidence support and entailment	92.4	90.8	91.6
Parameter conflict	97.1	95.6	96.3
Terminology inconsistency	93.5	91.2	92.3
Mandatory-element coverage	94.0	92.5	93.2
Cross-section inconsistency	96.3	94.9	95.6

As shown in Table 14, deterministic parameter and cross-section checks were more reliable than semantic entailment assessment. The verifier occasionally accepted paraphrases that omitted limiting conditions and occasionally rejected conservative claims

supported by several clauses rather than a single passage. Final system performance was therefore evaluated against expert annotations rather than the verifier's own decisions.

6.8 Expert assessment and operational efficiency

Document-quality assessment and evidence auditing were conducted as separate blinded tasks. Applicability and conflict labels achieved Fleiss' κ values of 0.81 (95% CI [0.77, 0.85]) and 0.78 (95% CI [0.72, 0.83]), respectively. Document ratings achieved ICC(2,k) = 0.84 for absolute agreement (95% CI [0.78, 0.89]). Table 15 reports expert ratings and operational time.

EC-ARAG received its highest expert rating for consistency and its smallest relative advantage for professional expression. LLM-only generation achieved a comparatively high expression score but weak regulatory correctness and consistency, demonstrating that linguistic fluency is not a reliable indicator of engineering validity. Compared with Closed-corpus Corrective RAG, EC-ARAG increased machine time by 29.7%, reduced human review time by 31.4%, and reduced total operational completion time by 21.4%.

Table 16 reports the average computational budget per complete document.

Table 16 confirms that EC-ARAG's quality gains were achieved at a higher agentic token and model-call cost. Evidence-token budgets remained similar, but EC-ARAG used substantially more intermediate verification and repair tokens.

6.9 Parameter sensitivity

Table 17 reports sensitivity to the final evidence size, and Table 18 reports sensitivity to the evidence-sufficiency threshold and repair iterations. Figure 8 integrates the three analyses.

Increasing top-k from 4 to 10 improved evidence coverage and applicability. Beyond 10, performance plateaued and then decreased. Raising the threshold from 0.78 to 0.90 improved evidence support by only 0.4 percentage points but increased machine time by 59.3%. Figure 8 also shows that the third repair iteration yielded only a 0.2-percentage-point support improvement.

6.10 Residual errors

The seed-42 document contained 55 unsafe claims and 26 condition-unverifiable claims. Three unsafe claims contained two distinct error types, producing 58 error instances. Table 19 and Figure 9 summarize the distribution.

Missing project conditions were the largest error source and were concentrated in special foundation treatment, drainage design, local elevation transitions, equipment-loading conditions, and temporary construction access. Figure 9 shows that roads, vertical planning, drainage, and foundation sections contained the highest missing-condition intensities.

6.11 Representative cases

Table 20 summarizes successful, abstained, and failed cases. Figure 10 presents their evidence-control sequences.

Table 15 Expert assessment and operational completion time.

Method	Applic. /5	Reg. correct. /5	Expression /5	Complete /5	Consistent /5	Editable /5	Machine min	Human review min	Total min
Template-based	3.62	3.71	3.69	3.58	3.77	3.41	1.8	82.4 ± 8.7	84.2 ± 8.7
LLM-only	3.19	3.08	4.02	3.44	3.21	3.28	3.2 ± 0.3	94.7 ± 9.6	97.9 ± 9.6
Dense RAG	3.73	3.68	4.08	3.76	3.64	3.52	4.8 ± 0.4	69.2 ± 7.4	74.0 ± 7.4
Hybrid RAG	4.06	4.03	4.20	4.09	4.02	3.96	5.7 ± 0.4	54.8 ± 6.1	60.5 ± 6.1
GraphRAG	4.12	4.09	4.18	4.14	4.16	4.07	7.4 ± 0.5	51.3 ± 5.7	58.7 ± 5.7
Closed-corpus Corrective RAG	4.21	4.19	4.24	4.22	4.23	4.17	9.1 ± 0.6	46.5 ± 5.2	55.6 ± 5.2
EC-ARAG	**4.55**	**4.58**	**4.43**	**4.51**	**4.61**	**4.46**	11.8 ± 0.7	31.9 ± 4.6	43.7 ± 4.8

Table 16 Average computational budget per complete document.

Method	Evidence tokens	Intermediate tokens	Output tokens	Mean LLM calls	Machine time (min)
LLM-only	0	9,800	9,420	6.0	3.2
Dense RAG	42,800	4,500	9,400	6.0	4.8
Hybrid RAG	44,100	4,800	9,350	6.0	5.7
GraphRAG	43,700	12,600	9,360	8.0	7.4
Closed-corpus Corrective RAG	44,300	24,900	9,310	14.2	9.1
EC-ARAG	45,200	61,800	9,280	32.4	11.8

Table 17 Sensitivity to final evidence size.

Final top-k	Applicability Macro-F1 (%)	Evidence support (%)	Machine time (min)
4	87.9	91.8	9.3
6	89.4	93.6	10.1
8	90.3	94.7	11.0
10	**90.6**	95.0	11.8
12	90.5	**95.1**	12.7
16	89.4	94.3	14.5

Table 18 Sensitivity to sufficiency threshold and repair iterations.

Setting	Evidence support (%)	Unsafe claims (%) ↓	Machine time (min)
Threshold 0.65	92.6	5.8	9.4
Threshold 0.72	94.0	4.6	10.3
Threshold 0.78	95.0	3.4	11.8
Threshold 0.84	95.3	3.2	14.6
Threshold 0.90	95.4	3.1	18.8
Zero repairs	89.0	8.7	8.4
One repair	93.1	5.2	10.2
Two repairs	95.0	3.4	11.8
Three repairs	95.2	3.2	13.4

Table 19 Residual error distribution.

Error type	Count	Percentage (%)
Missing project conditions leading to incorrect completion	18	31.0
Clause-parsing errors	11	19.0
Retrieval omissions	9	15.5
Conflict-resolution errors	7	12.1
Generation distortion	6	10.3
Coverage omissions	4	6.9
Incorrect memory update	3	5.2
Total error instances	58	100.0

The successful case demonstrates the interaction between applicability filtering and local repair. The abstention case demonstrates that conservative incompleteness may be safer than unsupported completion. The failure case shows that system reliability remains dependent on accurate regulatory parsing.

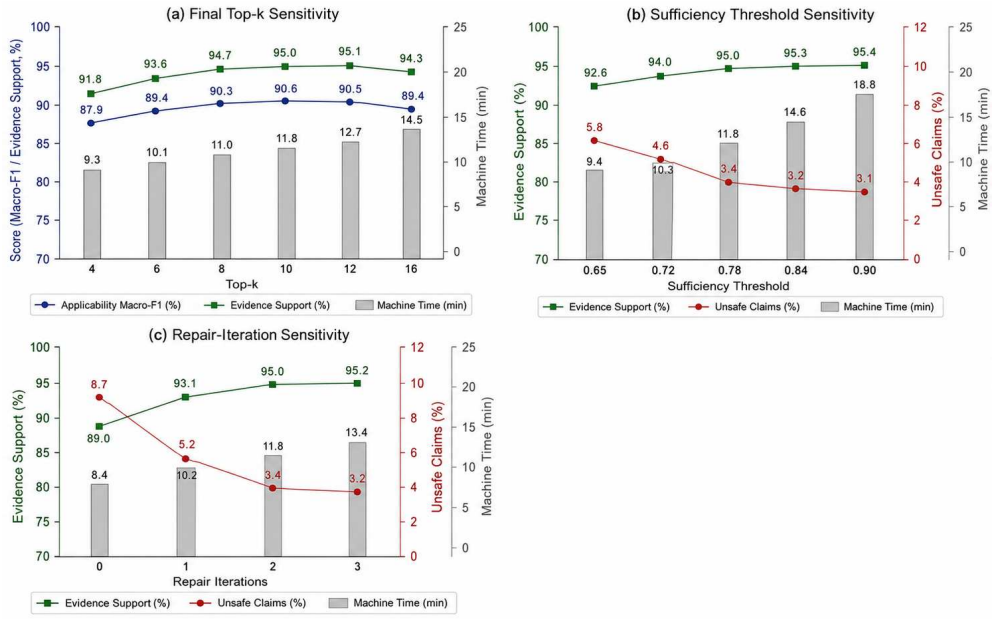


Fig. 8 Parameter sensitivity and quality–cost trade-offs. Panel (a) presents final top-k sensitivity, panel (b) presents evidence-sufficiency threshold sensitivity, and panel (c) presents repair-iteration sensitivity.

Table 20 Representative evidence-control cases.

Case	Evidence situation	System action	Final outcome
Structural-system success	Current regulation and typical design supported one option; historical text referred to incompatible conditions	Historical evidence was excluded and a missing applicability phrase was locally repaired	Accepted without factual correction
Drainage abstention	The regulation required elevation and discharge-boundary information absent from Project X inputs	The system avoided a definitive slope or discharge statement and created a confirmation item	Completed after the engineer supplied the condition
Foundation failure	A long clause note was incompletely parsed and an exception was omitted	The verifier failed to detect the missing exception in the final repair round	The expert replaced the affected statement

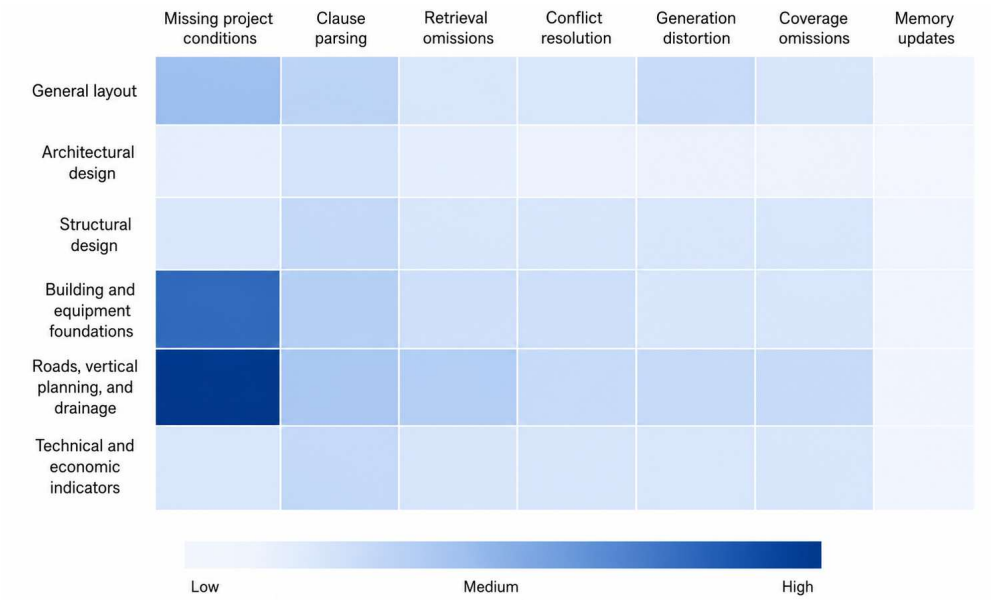


Fig. 9 Residual-error heat map by section and error type. Rows represent the six section groups, while columns represent missing conditions, clause parsing, retrieval omissions, conflict resolution, generation distortion, coverage omissions, and memory updates.

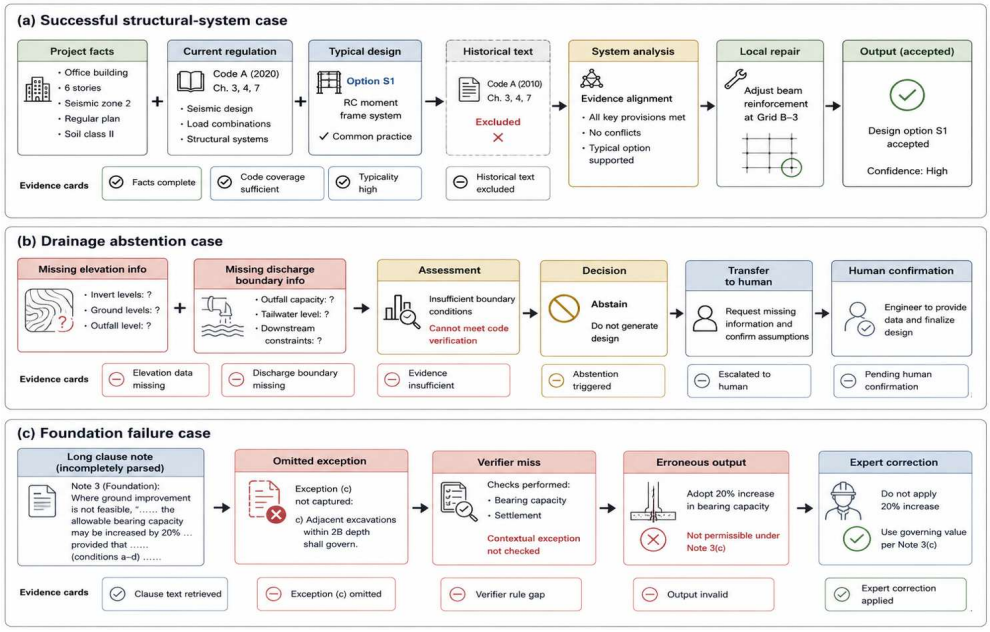


Fig. 10 Successful resolution, abstention, and residual failure. Panel (a) presents the structural-system evidence chain and local repair. Panel (b) presents a drainage proposition transferred to human confirmation because of missing conditions. Panel (c) presents a foundation exception omitted during parsing and subsequently corrected by an expert.

7 Discussion

7.1 Regulation-aware representation improves evidence validity

The largest retrieval-related improvement occurred in applicability Macro-F1. Removing applicability modelling reduced Macro-F1 by 6.5 percentage points and increased the unsafe-claim rate from 3.4% to 7.0%. These findings indicate that an engineering requirement cannot be represented solely as a semantically searchable passage. A numerical requirement must remain linked to its object, conditions, unit, exception, modality, authority, and version. Otherwise, a retriever may return the correct numerical value under an incorrect engineering condition.

Validity gating was also important. Obsolete sources, discipline mismatches, and hard-condition contradictions were excluded before ranking. A high semantic-similarity score could therefore not compensate for regulatory invalidity. The findings extend previous engineering RAG research on hybrid indexing, hierarchical knowledge pools, explicit engineering facts, and domain rules [14, 15, 19, 23].

7.2 Dual-loop control addresses evidence insufficiency and misuse

The planning–retrieval and generation–verification–repair loops addressed different failure modes. The planning–retrieval loop improved coverage by decomposing broad section headings into independently retrievable propositions. The repair loop affected output quality after retrieval. Removing it increased the unsafe-claim rate from 3.4% to 8.7%, although applicability Macro-F1 remained nearly unchanged. Correct evidence may therefore still be distorted through omitted conditions, altered values, unsupported generalization, or inconsistent terminology.

Local repair was preferable to complete section regeneration because it retained verified content and reduced opportunities for introducing new errors. The sensitivity results further show that two repair iterations captured most of the available benefit.

7.3 Evidence sources must be assigned distinct roles

Regulations, project facts, typical designs, and historical documents are not institutionally equivalent. Historical documents may be professionally written but contain obsolete requirements or project-specific parameters. Typical designs may provide recommended configurations but not binding conclusions. Project inputs define the current context but cannot override mandatory constraints.

Removing source-role prioritization reduced conflict-resolution F1 from 91.8% to 84.2%. Removing the complete conflict-resolution agent reduced it further to 75.6%, while completeness remained above 95%. A document may therefore be comprehensive in appearance but unsafe in evidence composition. The abstention mechanism also contributed to reliability: 26 seed-42 claims were not converted into definitive conclusions because essential conditions were unavailable.

7.4 Project-state memory makes drafting a document-level task

The largest deterioration in consistency occurred when global project-state memory was removed. Parameter consistency decreased from 97.0% to 88.6%. This finding confirms that preliminary design drafting is not a sequence of independent question-answering tasks. Once a building name, floor count, structural system, foundation type, road parameter, or area has been verified, it becomes a constraint on subsequent sections.

Memory also creates propagation risk. An incorrect early write may affect several sections. The confirmed–derived–provisional–rejected state model and deterministic write gate limited this risk, although three incorrect memory writes remained.

7.5 Evidence extraction and engineering efficiency

The extraction experiment identified applicability conditions and exceptions as the most difficult fields. These fields are also among the most important for safe engineering interpretation. Retrieval and generation cannot recover an applicability condition that was not represented correctly in the evidence unit.

The one-time construction of the evidence base required 31.2 person-hours of review. EC-ARAG is consequently most economically appropriate when the resulting knowledge base is reused across multiple projects. The reported operational-time reduction should not be interpreted as eliminating the initial cost of domain-knowledge preparation.

EC-ARAG increased machine inference time but reduced total operational time. Compared with Closed-corpus Corrective RAG, machine time increased by 29.7%, whereas professional review time decreased by 31.4%. Total operational completion time decreased by 21.4%. The computational-budget analysis also shows that EC-ARAG used more intermediate tokens and LLM calls. Its quality advantage is therefore associated with a transparent resource cost rather than being computationally free.

7.6 Negative findings, oversight, and limitations

The experiments produced several non-ideal findings. Roads, vertical planning, drainage, and foundations remained more difficult than architectural design because text-only inputs did not fully represent geometry, elevation, loading, or detailed geotechnical conditions. EC-ARAG's relative advantage in professional expression was smaller than its advantage in regulatory correctness; evidence-constrained drafting was safer but occasionally repetitive. Stricter evidence thresholds increased latency and abstention. Exception parsing remained imperfect, and the verifier did not achieve perfect entailment classification.

EC-ARAG is a drafting and verification assistant rather than an autonomous engineering approval system. Structural-system selection, foundation design, unusual site treatment, and conclusions depending on incomplete geotechnical or geometric information must be reviewed by qualified professionals. The system records source identifiers, versions, applicability conditions, selected evidence, generated claims, verification decisions, repairs, state updates, and unresolved items, but this traceability does not transfer engineering responsibility to the software.

The study is limited to a single project. Five random seeds characterize stochastic stability but do not represent five independent substation projects. The method cannot

verify consistency against drawings, BIM models, topographic surveys, or equipment-layout geometry. Evidence quality depends on clause parsing and standard-version maintenance. The knowledge base required non-trivial construction effort, and the experiments used one model family and one organizational corpus.

8 Conclusion

This study proposed EC-ARAG, an evidence-constrained agentic retrieval-augmented generation framework for drafting substation civil engineering preliminary design documents.

For RQ1, regulation-aware evidence representation and validity-gated retrieval achieved a Recall@5 of 94.1% and an applicability Macro-F1 of 90.6%. For RQ2, the dual planning–retrieval and generation–verification–repair architecture achieved an evidence support rate of 95.0% and section completeness of 96.0%. Removing the repair loop increased the unsafe-claim rate from 3.4% to 8.7%. For RQ3, source-role-aware conflict resolution achieved a conflict-resolution F1 of 91.8%, while global project-state memory supported parameter consistency of 97.0%; removing the memory reduced consistency to 88.6%. For RQ4, EC-ARAG increased machine inference time to 11.8 min but reduced professional review time to 31.9 min. Total operational completion time was 43.7 min, 21.4% lower than that of Closed-corpus Corrective RAG.

The findings indicate that regulation-intensive document drafting depends not only on retrieval recall but also on evidence applicability, source authority, conflict handling, claim verification, abstention, and document-level state control. The results should not be interpreted as evidence of autonomous engineering approval or as proof of generalization to all substation projects. Future work will integrate textual descriptions with drawings and BIM information, improve the extraction of regulatory tables and exceptions, support automatic monitoring of standard versions, and evaluate the framework across multiple substation projects and engineering organizations.

Supplementary information. The implementation code, prompt templates, configuration files, random seeds, software-dependency specification, baseline settings, evaluation scripts, and figure-generation scripts are submitted as supplementary materials accompanying the manuscript.

Declarations

Funding. This work was supported by the Science and Technology Project of State Grid Shanghai Electric Power Company under Grant No. 52090R250002.

Competing interests. The authors declare that they have no competing interests.

Ethics approval and consent to participate. This study involved human participants solely in the expert evaluation of generated preliminary design documents. The expert-evaluation protocol was reviewed and approved by the Ethics Committee of State Grid Shanghai Electric Power Design Co., Ltd. before the evaluation began. The study was conducted in accordance with the Declaration of Helsinki and the applicable guidelines and regulations of the approving committee for research involving human

participants. All participating experts received a detailed explanation of the study objectives and procedures and provided written informed consent before participation. Participation was voluntary, and no personally identifiable information was collected or reported.

Consent for publication. Not applicable. The manuscript contains no identifiable images or other identifiable personal information.

Data availability. The data used in this study comprise national and industry standards, enterprise standards, typical-design documents, historical substation civil engineering preliminary design documents, anonymized project-input documents, manually annotated retrieval queries, regulatory-applicability cases, evidence-conflict cases, and expert-evaluation records. No public benchmark dataset was used; therefore, no public dataset download link is applicable. The full texts of certain standards, enterprise documents, historical engineering documents, and project-specific materials cannot be publicly redistributed because of copyright, confidentiality, and organizational-access restrictions. Anonymized project facts, retrieval-query sets, evidence metadata, applicability and conflict annotations, claim-level evaluation labels, aggregated results, and other derived data supporting the findings are available from the corresponding author upon reasonable request.

Code availability. The source code associated with this study, including corpus preprocessing, regulation-aware evidence construction, dense and sparse retrieval, graph-based evidence expansion, reranking, agentic RAG orchestration, conflict resolution, project-state memory management, claim-level verification and repair, metric calculation, statistical analysis, and result visualization, is provided as part of the supplementary materials accompanying this article. The supplementary package also includes configuration files, prompt templates, software dependencies, random seeds, and reproduction instructions.

Author contributions. Y.H.: Conceptualization, methodology, supervision, project administration, resources, funding acquisition, and writing–original draft and writing–review and editing. X.Z.: Software, data curation, investigation, validation, formal analysis, visualization, and writing–original draft. Both authors contributed to the interpretation of the results and approved the final manuscript.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., Yih, W.-t., Rocktaschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474 (2020)

[2] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H.: Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997 (2023)

[3] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models. In: Proceedings of the 11th International Conference on Learning Representations (2023)

[4] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 36, pp. 8634–8652 (2023)

[5] Jiang, Z., Xu, F.F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., Neubig, G.: Active retrieval augmented generation. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 7969–7992 (2023). https://doi.org/10.18653/v1/2023.emnlp-main.495

[6] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-refine: Iterative refinement with self-feedback. In: Advances in Neural Information Processing Systems, vol. 36, pp. 46534–46594 (2023)

[7] Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-rag: Learning to retrieve, generate, and critique through self-reflection. In: Proceedings of the 12th International Conference on Learning Representations (2024)

[8] Yan, S.-Q., Gu, J.-C., Zhu, Y., Ling, Z.-H.: Corrective retrieval augmented generation. arXiv preprint arXiv:2401.15884 (2024)

[9] Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2024)

[10] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Larson, J.: From local to global: A graphrag approach to query-focused summarization. arXiv preprint arXiv:2404.16130 (2024)

[11] Es, S., James, J., Espinosa-Anke, L., Schockaert, S.: Ragas: Automated evaluation of retrieval augmented generation. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pp. 150–158 (2024)

[12] Saad-Falcon, J., Khattab, O., Potts, C., Zaharia, M.: Ares: An automated evaluation framework for retrieval-augmented generation systems. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 338–354 (2024). https://doi.org/10.18653/v1/2024.naacl-long.20

[13] Wang, X., Wang, Z., Gao, X., Zhang, F., Wu, Y., Xu, Z., Shi, T., Wang, Z., Li, S., Qian, Q., Yin, R., Lv, C., Zheng, X., Huang, X.: Searching for best practices

in retrieval-augmented generation. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 17716–17736 (2024). https://doi.org/10.18653/v1/2024.emnlp-main.981

[14] Wang, Z., Liu, Z., Lu, W., Jia, L.: Improving knowledge management in building engineering with hybrid retrieval-augmented generation framework. Journal of Building Engineering **103**, 112189 (2025) https://doi.org/10.1016/j.jobe.2025.112189

[15] Wu, C., Ding, W., Jin, Q., Jiang, J., Jiang, R., Xiao, Q., Liao, L., Li, X.: Retrieval augmented generation-driven information retrieval and question answering in construction management. Advanced Engineering Informatics **65**, 103158 (2025) https://doi.org/10.1016/j.aei.2025.103158

[16] Lee, J., Ahn, S., Kim, D., Kim, D.: Performance comparison of retrieval-augmented generation and fine-tuned large language models for construction safety management knowledge retrieval. Automation in Construction **168**, 105846 (2024) https://doi.org/10.1016/j.autcon.2024.105846

[17] Uhm, M., Kim, J., Ahn, S., Jeong, H., Kim, H.: Effectiveness of retrieval augmented generation-based large language models for generating construction safety information. Automation in Construction **170**, 105926 (2025) https://doi.org/10.1016/j.autcon.2024.105926

[18] Baek, S., Park, C.Y., Jung, W.: Automated safety risk management guidance enhanced by retrieval-augmented large language model. Automation in Construction **176**, 106255 (2025) https://doi.org/10.1016/j.autcon.2025.106255

[19] Siddharth, L., Luo, J.: Retrieval augmented generation using engineering design knowledge. Knowledge-Based Systems **303**, 112410 (2024) https://doi.org/10.1016/j.knosys.2024.112410

[20] Yang, F., Zhang, J.: Prompt-based automation of building code information transformation for compliance checking. Automation in Construction **168**, 105817 (2024) https://doi.org/10.1016/j.autcon.2024.105817

[21] Xue, X., Zhang, J., Chen, Y.: Question-answering framework for building codes using fine-tuned and distilled pre-trained transformer models. Automation in Construction **168**, 105730 (2024) https://doi.org/10.1016/j.autcon.2024.105730

[22] Shi, J.W.L., Solihin, W., Yeoh, J.K.W.: Fine-tuning a large language model for automated code compliance of building regulations. Advanced Engineering Informatics **68**, 103676 (2025) https://doi.org/10.1016/j.aei.2025.103676

[23] Xiong, X., Cai, H., Yu, H., Shen, B., Hu, P.: Dr-rag: Domain-rule-based retrieval-augmented generation for aviation digital model design. Advanced Engineering Informatics **68**, 103688 (2025) https://doi.org/10.1016/j.aei.2025.103688

[24] Chen, J., Bao, Y.: Multi-agent large language model framework for code-compliant automated design of reinforced concrete structures. Automation in Construction **177**, 106331 (2025) https://doi.org/10.1016/j.autcon.2025.106331

[25] Qian, Y., Han, B., Yuan, Y., Zhang, X., Zhu, H., Ni, L., *et al.*: Hierarchical multi-agent reinforcement learning for retrieval-augmented industrial document question answering. Scientific Reports **16**, 13512 (2026) https://doi.org/10.1038/s41598-026-41684-z

[26] Qian, C., Chen, Y.-Y., Yang, J.-Y., Le, X.-W., Shen, X.-Y., Zeng, Y.-H.: Knowledge graph-enhanced framework for electric power engineering report generation. Journal of Electronic Science and Technology **24**(1), 100344 (2026) https://doi.org/10.1016/j.jnlest.2025.100344

[27] Zhou, L., Gao, Z.-T., Zhai, P.-F., Zeng, Y.-H.: Intelligent parameter recommendation for substation design using knowledge graph and graph neural networks. Energy Informatics **9**, 11 (2026) https://doi.org/10.1186/s42162-025-00605-6

[28] Tang, W., Zhang, Y., Mao, X., Jia, H., Lv, K., Shan, L., Qiao, Y., Jiang, T.: Construction and application of knowledge graph for power grid new equipment start-up. Energies **18**(20), 5471 (2025) https://doi.org/10.3390/en18205471

[29] Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval **3**(4), 333–389 (2009)

[30] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: Bge m3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In: Findings of the Association for Computational Linguistics: ACL 2024, pp. 10024–10034 (2024)

[31] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pp. 3982–3992 (2019)

[32] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-t.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 6769–6781 (2020)

[33] Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)

[34] Yang, A., et al.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2025)

[35] Knollmeyer, S., Caymazer, O., Grossmann, D.: Document graphrag: Knowledge graph enhanced retrieval augmented generation for document question answering within the manufacturing domain. Electronics **14**(11), 2102 (2025) https://doi.

org/10.3390/electronics14112102

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ECARAGCompleteCodeProject.zip