electronics

MDPI

Article

HG-RAG: Hierarchical Graph-Enhanced Retrieval-Augmented Generation for Power Systems

Zhijun Shen [1], Xinlei Cai [1], Binye Ni [1], Zijie Meng [1], Zhanhong Huang [2,*] and Tao Yu [2]

[1] Power Dispatch Control Department, Power Dispatch Control Center of Guangdong Power Grid Co., Ltd., Guangzhou 510699, China
[2] School of Power Engineering, College of Electrical Engineering, South China University of Technology, Guangzhou 510640, China
* Correspondence: 202211082838@mail.scut.edu.cn; Tel.: +86-13680462543

Abstract

Retrieval-augmented generation (RAG) has shown strong potential for knowledge-intensive tasks, yet its performance degrades sharply when applied to structured long-context documents in power systems, where dense entity–relation dependencies, cross-document references, and strict traceability requirements exist. To address this Structured Long-Context RAG (SLCRAG) challenge, this paper proposes a hierarchical graph-enhanced RAG (HG-RAG) framework tailored for power system question answering. HG-RAG constructs a globally consistent knowledge graph via sliding-window entity–relation extraction to mitigate semantic fragmentation, and employs multi-granularity structured indexing for precise entity/relation retrieval. A hierarchical structured retrieval mechanism with multi-hop expansion and semantic distillation maximizes recall while minimizing redundancy. Furthermore, a regex-enhanced retrieval module records authoritative file_path provenance and constrains downstream retrieval to the same source documents, effectively eliminating cross-document interference—especially in cases where different documents contain similar entities and relations. Combined with version control and deduplication-merging, HG-RAG supports incremental knowledge updates with minimal forgetting and negligible token overhead. Experimental results on a domain-authentic power system QA dataset demonstrate that HG-RAG outperforms LightRAG and GraphRAG, achieving up to 85.47% accuracy in short-answer tasks with significantly lower token consumption. Ablation studies confirm that semantic distillation primarily improves precision and efficiency, while regex-enhanced retrieval safeguards recall in edge cases.

Keywords: structured long-context RAG; power system dispatching; knowledge graph; hierarchical retrieval; regex-enhanced retrieval; incremental update

Check for updates

Academic Editor: Jen-Hao Teng

Received: 6 March 2026
Revised: 23 March 2026
Accepted: 28 March 2026
Published: 30 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Power system dispatching is a knowledge-intensive task that involves executing safety-critical analyses and decisions based on heterogeneous rules, mechanisms, and operational constraints. Over years of operational practice, dispatching systems have accumulated extensive knowledge artifacts, including structured monitoring data, procedural documents, and historical dispatch logs. For complex dispatching tasks, operational procedures require the rapid identification and cross-analysis of heterogeneous knowledge sources under temporal constraints. Dispatch strategies heavily rely on the integration and interpretation of cross-textual knowledge, where incomplete or mismatched knowledge retrieval may directly lead to state misjudgments and human-factor risks in decision-making. Consequently,

achieving fast, accurate, and traceable retrieval of distributed dispatching knowledge has emerged as a critical challenge for real-time analytical efficiency in online dispatch systems.

Existing dispatching knowledge systems exhibit high structural complexity and dispersion, yet lack standardized tagging. Dispatchers currently rely on keyword-based identification to retrieve knowledge clauses and query data, a process prone to partiality and operational risks. Current challenges in dispatching knowledge retrieval and utilization include: (1) Fragmented cross-document organization, which significantly undermines semantic matching and knowledge accessibility; (2) Contextual discrepancies and ambiguous mappings between documents, complicating precise knowledge association and invocation; (3) Semantic ambiguity of labels for physical devices and technical terminology, resulting in information misalignment. To address these limitations, there is an urgent need for a novel retrieval framework integrating robust entity recognition and hierarchical knowledge organization.

In recent years, large language models (LLMs) have demonstrated remarkable advancement in natural language understanding and generation. Among them, the widely used retrieval-augmented generation (RAG) endows LLMs with the ability to embed knowledge through retrieval reasoning of external knowledge and dynamic injection into the generation context. Recent advancements in RAG increasingly focus on targeted enhancements across the retrieval–generation pipeline, including improvements in retrieval accuracy, context integration, and response generation [1]. For long-context and structured document applications, RAG enhances the accuracy of LLMs by retrieving information from both unstructured and structured data sources, though performance can degrade when cross-document dependencies are not properly modeled [2]. While some initial efforts have explored RAG for domain-specific knowledge in power systems, such as using retrieval-augmented methods for knowledge-assisted generation of standard documents [3], these approaches often struggle with the dense entity–relationship dependencies, multi-modal references, and intricate cross-references typical in dispatching scenarios. Directly applying existing RAG methods to such structured long-context documents leads to significant performance decline, including cross-document association semantic bias and weak generalization of incremental knowledge. We refer to this challenge as the Structured Long-Context RAG (SLCRAG) problem, defined as retrieving and generating responses that require reasoning across multiple document segments while ensuring high precision, recall, and traceability.

In response to the capability requirements of cross-document aggregation, associative logical reasoning, and finite-domain adaptive semantic matching for SLCRAG tasks in the field of power dispatching, we propose a hierarchical graph-enhanced RAG (HG-RAG) algorithm with traceability and generalization. This improved architecture is specifically designed for enhancing the overall semantic understanding capabilities of long and cross-document structured documents in the field of power dispatching.

The main contributions of this work are as follows:

(1) We defined the SLCRAG problem for the power field and innovatively proposed the hierarchical graph-enhanced RAG (HG-RAG) framework to address the common query requirements of heterogeneous knowledge in power dispatching.
(2) A dynamic context sliding-window semantic extraction mechanism integrating structured knowledge graphs is proposed to support the aggregated representation of cross-module information and the streaming reasoning of graph context.
(3) A structured retrieval algorithm design based on graph multi-hop retrieval and token label is proposed to enhance the cross-dependency mining of semantic entities among multiple documents and the filtering of redundant information.

https://doi.org/10.3390/electronics15071445

(4) A traceable regex-enhanced rollback retrieval mechanism and retrieval mode were proposed. By matching approximate label knowledge modules to dynamically expand the graph and quantify the dynamic adaptability of knowledge, the robustness for edge situations and irregular query scenarios can be addressed.

The remainder of the paper is structured as follows. Section 2 introduces the description and modeling of SLCRAG problems in the field of power dispatching; Section 3 presents the formulation for HG-RAG; Section 4 presents the LLM-based reasoning module in HG-RAG; then, case studies and experimental results are discussed in Sections 5 and 6, followed by the discussion in Section 7 and the conclusions in Section 8.

2. Background and Related Work

2.1. Background on Retrieval-Augmented Generation (RAG)

Retrieval-augmented generation (RAG) has emerged as a powerful paradigm for knowledge-intensive tasks such as open-domain QA and legal reasoning [4], by dynamically incorporating external knowledge into the generation stage to improve factual consistency and contextual relevance. A conventional RAG framework typically consists of three steps: (i) document chunking and index construction, (ii) retrieval of relevant content based on a query, and (iii) answer generation by fusing the query with the retrieved content. The overall workflow is illustrated in Figure 1. In this process, the retrieved knowledge serves as additional factual and contextual support for the large language model (LLM).

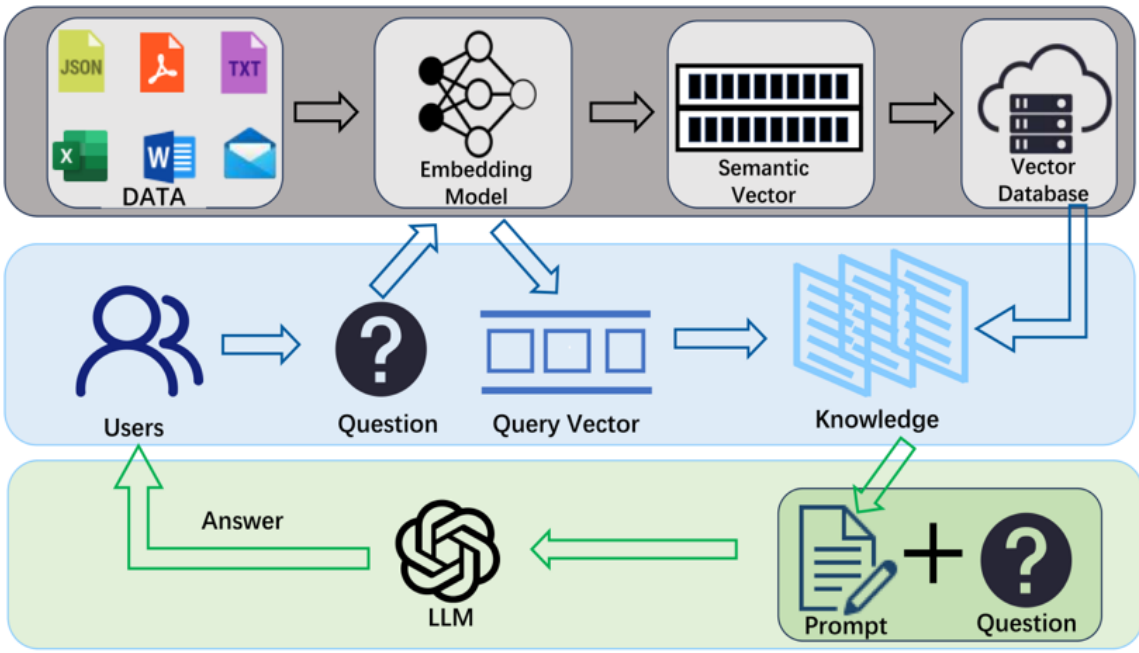


Figure 1. Overall architecture of a conventional RAG system.

For theoretical analysis, a RAG system can be formalized as:

$$M = G(q, \psi(q; D)),\ \psi(q; D) = R(q; \widehat{D}),\ \widehat{D} = \varphi(D) \tag{1}$$

where M denotes the entire RAG system, G is the generation module, and ψ is the retrieval module, which consists of an indexer φ and a retriever R. The indexer φ processes the original document collection D (e.g., chunking and embedding) to produce the indexed structure $\widehat{D}$. The retriever R searches the index $\widehat{D}$ for content relevant to the user query q. The generation module G then produces the final answer by conditioning on both the original query q and the retrieved content $\psi(q; D)$.

2.2. Problem Formulation of SLCRAG

Let $\mathcal{D} = \{d_i\}_{i=1}^{N}$ denote a corpus of structured power-system documents and $\mathcal{C} = \{c_j\}_{j=1}^{M}$ the chunk set derived from $\mathcal{D}$. Given a user query q, the Structured Long-Context Retrieval-Augmented Generation (SLCRAG) task aims to retrieve an evidence subset $\mathcal{S}(q) \subseteq \mathcal{C}$ and generate an answer a.

Different from conventional RAG, SLCRAG must satisfy the following requirements:

1. Evidence support: every factual claim in a should be supported by at least one chunk in $\mathcal{S}(q)$;
2. Dependency preservation: if answering q requires cross-chunk or cross-document reasoning, the corresponding dependency chain should be preserved in $\mathcal{S}(q)$;
3. Traceability: each retrieved chunk should be associated with provenance metadata for source verification;
4. Token efficiency: the total length of the retrieved evidence should satisfy the token budget constraint.

Formally, the token budget constraint is written as

$$\sum_{c \in \mathcal{S}(q)} \text{len}(c) \leq B \tag{2}$$

where B denotes the maximum context budget for answer generation.

Accordingly, SLCRAG can be formulated as a constrained retrieval-generation problem that jointly balances completeness, faithfulness, traceability, and efficiency for structured long-context document question answering.

2.3. Existing Methods and Limitations

Under the above formulation, an effective SLCRAG solution should simultaneously satisfy four key requirements: evidence completeness, dependency preservation, source-level traceability, and token efficiency. However, existing RAG methods are not specifically designed for structured long-context power-system documents and therefore cannot satisfy these requirements in a unified manner. Regulatory standards, operation manuals, and market rules are typically lengthy, highly structured, and rich in domain-specific terminology and entity–relation dependencies [1]. They also contain frequent cross-clause and cross-table references. As a result, directly applying generic retrieval-augmented generation (RAG) pipelines often leads to semantic fragmentation caused by block-wise chunking, incomplete recall of cross-block dependencies, and weak adaptation to business-specific labels such as voltage levels, dispatch hierarchies, and equipment categories. Furthermore, the stringent compliance requirements of the power industry demand fine-grained traceability of retrieved evidence, which conventional RAG systems generally fail to guarantee.

To address the limitations of standard RAG for complex tasks, prior studies have explored structured knowledge integration and graph-enhanced retrieval. General surveys on knowledge graphs and their applications summarize major advances in knowledge representation and reasoning [2], while recent surveys on RAG for AI-generated content further review efficiency-oriented and integration-oriented techniques [3]. Foundational RAG frameworks for knowledge-intensive tasks establish general paradigms for incorporating external knowledge sources [4], and subsequent studies on unifying large language models with knowledge graphs provide systematic directions for structured knowledge integration [5]. Complementary approaches such as KGLM incorporate knowledge graph structure into language model pre-training [6], while confidence-aware graph neural networks demonstrate the adaptation of graph-based reasoning to power system reliability assessment [7]. Recent domain-oriented studies such as PowerGraph-LLM [8],

feedback-driven multi-agent frameworks for power system simulation [9], multi-agent deep constrained Q-learning for smart building energy management [10], and dispatching automation systems integrating LLMs, KGs, and AI agents [11] further extend structured reasoning and decision support in power-related scenarios. Nevertheless, these methods mainly improve representation, simulation, or operational intelligence at a general level, and do not explicitly address dependency restoration across multiple document blocks under the compliance constraints of power-system documentation.

A distinct research trajectory strengthens complex reasoning through multi-hop retrieval, graph traversal, and agent-based planning. Knowledge graph prompting for multi-document question answering formulates retrieval as iterative graph traversal [12], and Think-on-Graph further improves multi-hop reasoning by combining search and pruning strategies [13]. Related studies in distribution system restoration demonstrate the value of collaborative reasoning and dynamic reconfiguration in complex operational settings [14], while HopRAG advances logic-aware multi-hop retrieval for retrieval-augmented generation [15]. Multi-source RAG-LLM frameworks for sustainable energy transitions further reveal the benefits and limitations of heterogeneous evidence fusion [16], and security analyses of large language models in power systems emphasize the importance of structured planning for reducing reasoning drift and operational risks [17]. In addition, multi-agent reinforcement learning for critical load restoration [18] and structured reasoning frameworks for complex logical reasoning and knowledge planning [19] both highlight the importance of decomposition, coordination, and verification in challenging decision environments. However, although these methods are effective for explicit reasoning and retrieval planning, they often incur higher inference latency and remain insufficiently adapted to power-system operational rules, label taxonomies, and provenance-sensitive document usage.

Within the energy and power systems sectors specifically, tailored RAG applications have begun to emerge. MGRAG addresses multimodal work ticket processing in power grids [20], and IHGR-RAG supports equipment condition assessment through enhanced retrieval-augmented generation [21]. Recent studies have also investigated electric-power QA with retrieval augmentation and KG-enhanced LLMs. A dedicated RAG framework for electric power industry question answering improves retrieval through hierarchical semantic segmentation and hybrid retrieval [22], while KG-enhanced prompting [23] and a self-consistency, extract-and-rectify framework [24] further improve the reliability of electric-power QA by incorporating structured knowledge. These studies demonstrate the growing feasibility of domain-adapted RAG in power applications, but their target scenarios remain centered on multimodal ticket handling and equipment assessment rather than structured long-context question answering over compliance-intensive dispatching documents. In particular, they do not jointly address cross-block dependency restoration, provenance-aware evidence organization, and token-efficient context compression under the SLCRAG setting.

An additional research stream aims to improve explainability and retrieval optimization. Recent work has introduced interactive multimodal frameworks that integrate retrieval-augmented generation with visualization capabilities [25]. QA-GNN constructs joint graphs linking question-answering context with knowledge graph entities to reduce irrelevant node interference and strengthen valid dependency utilization [26]. LightRAG and GraphRAG represent recent advances in efficient retrieval-augmented generation and graph-based knowledge management [27,28]. Although these methods improve transparency and retrieval organization, in power systems they rarely incorporate constraints such as permission rules, operational dependencies, or business label encoding, resulting in persistent compliance and precision gaps. Compared with recent graph-enhanced and multi-hop RAG frameworks, HG-RAG does not use graph structure only for generic

traversal or memory organization; instead, it builds a provenance-aware document graph tailored to dispatching regulations and couples sliding-window dependency restoration, regex-enhanced source anchoring, bounded multi-hop retrieval, and semantic distillation to jointly preserve evidence completeness, source-level traceability, and token efficiency. This emphasis on robustness and traceable evidence organization is also consistent with resilience-oriented research in power systems, where maintaining reliable operation under partial measurements, communication disturbances, and cross-layer attacks is a core design concern [29,30]. These limitations motivate the need for a retrieval-generation framework specifically designed for structured long-context power-system documents.

3. Formulation for HG-RAG

To address the above gap, we propose HG-RAG, a hierarchical graph-based retrieval-augmented generation framework for structured long-context question answering in power-system documents. Existing representative methods in the power domain focus on different task settings, including graph-based power grid reasoning and optimization [8], multi-modal work ticket processing [20], and equipment condition assessment [21]. In contrast, HG-RAG is designed for compliance-oriented document question answering, where cross-block dependency preservation, source-level traceability, and token-efficient evidence organization must be considered jointly under the SLCRAG setting.

As discussed in Section 2, conventional RAG pipelines (Figure 1) provide a solid foundation for integrating retrieval into LLM-based QA, but their direct application to power system documentation faces several domain-specific challenges:

1. **Highly structured, long-form technical documents.** Power system regulations, operation manuals, and market rules are lengthy, richly structured, and contain complex cross-references. Naive chunking often fragments semantics, splitting key entities and relationships across chunk boundaries and thereby losing critical context during extraction.
2. **Stringent accuracy and traceability requirements.** Industrial applications require retrieval mechanisms capable of handling both detailed, entity-specific queries and high-level, abstract questions while ensuring complete transparency in the reasoning process.
3. **Token efficiency constraints.** The large volume of technical content risks excessive token usage, which increases computational cost and latency.

To address these challenges, we propose HG-RAG, an enhanced RAG framework tailored for power system document QA. The overall architecture is shown in Figure 2.

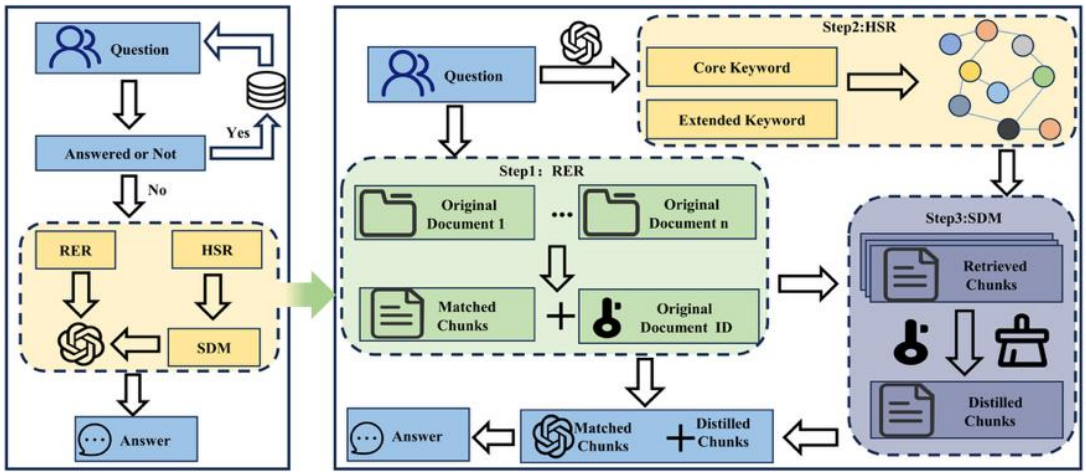


Figure 2. HG-RAG framework.

Building upon the standard RAG paradigm, HG-RAG introduces three core innovations:

1. **Sliding-window entity–relationship extraction with knowledge graph construction.** To alleviate semantic loss caused by direct fixed-length truncation, an overlapping sliding-window strategy is adopted during document segmentation. With overlapping neighboring chunks, content that **would** otherwise be split at chunk boundaries can still be preserved in adjacent chunks, thereby improving the continuity of sentence-level and local semantic information. Based on these overlapping chunks, HG-RAG extracts entities and relationships to construct a globally consistent structured knowledge graph, substantially improving the integrity and coverage of structured evidence.
2. **Structured retrieval with multi-hop expansion and semantic distillation.** Leveraging the constructed knowledge graph, HG-RAG performs precise structured retrieval and augments it with multi-hop expansion to recover cross-clause dependencies. A semantic distillation module then filters redundant or weakly relevant content, ensuring comprehensive recall while significantly reducing token consumption.
3. **Regex-enhanced retrieval.** For abstract or implicit queries, as well as cases where the knowledge graph provides insufficient coverage, a regular-expression-based block-level retrieval mechanism is employed as a strategy. This approach addresses knowledge blind spots and irregular expressions, thereby enhancing robustness in complex and boundary cases.

Given a user query, HG-RAG first executes the Regex-Enhanced Retrieval (RER) module to search for exact matches in the original document. Simultaneously, Hierarchical Structured Retrieval (HSR) decomposes the query into core keywords and expanded keywords to retrieve relevant text blocks from the structured knowledge graph. HER results are first cleansed with RER matching results to ensure that candidate text blocks come from the same original document. They are then processed by the Semantic Distillation Module (SDM) to filter out redundant information. Finally, the distilled context is input into the Regular Expression Module (LLM) to generate traceable and accurate answers.

These components collectively form the core of HG-RAG's enhanced RAG module, enabling robust, efficient, and accurate knowledge retrieval for power system QA. The subsequent subsections provide a detailed description of each component.

3.1. Graph-Enhanced Entity and Relationship Extraction via Sliding Windows

The sliding-window entity–relationship extraction module in HG-RAG is designed to reduce semantic loss caused by direct fixed-length truncation. If a document is segmented only by a hard token limit, the boundary may fall in the middle of a sentence or a semantically continuous passage, which may break the original meaning and lead to incomplete context preservation. To address this issue, HG-RAG adopts an overlapping sliding-window strategy during document segmentation.

Formally, given an ordered token sequence $D = \{t_1, t_2, \ldots, t_N\}$, the chunk size w denotes the number of tokens contained in each chunk, and the sliding step size s denotes the shift between the starting positions of two adjacent chunks. The k-th chunk is defined as

$$C_k = \{t_{(k-1)s+1}, \ldots, t_{min((k-1)s+w,N)}\} \tag{3}$$

In this work, the chunk size was set to $w = 1200$ tokens and the sliding step size was set to $s = 1000$ tokens. This setting allows adjacent chunks to share local context, so that sentence-level and locally continuous semantics near chunk boundaries can still be preserved, while keeping redundancy and computational overhead at a manageable level. The resulting overlapping chunks serve as semantic units for subsequent entity–relationship

extraction and knowledge graph construction, thereby improving both the integrity and coverage of structured evidence.

In the context of power dispatching operations, the constructed knowledge graph incorporates multiple categories of semantic objects:

1. **Equipment entities:** This category includes primary equipment (e.g., main transformer, circuit breaker, disconnect switch), secondary equipment (e.g., protective relay, dispatch automation system), and auxiliary equipment (e.g., capacitor bank, backup power supply).
2. **Organization and role entities:** Examples include the provincial dispatch center, regional control center, on-duty operator, and maintenance crew.
3. **Regulation and clause entities:** This category covers specific clauses from the Power Dispatching Regulation (e.g., Safety Code Article 4.2), operational procedures (e.g., forced energization operation), and dispatch command types (e.g., isolation command, parallel connection command).
4. **State and condition entities:** Examples include maintenance state, emergency shutdown, busbar energized, and system frequency low.
5. **Temporal and event entities:** Examples include a trip event on "Aug 15, 2023, 14:35" and the effective time of a dispatch maintenance plan.

For each sliding window, the LLM-based extraction module $\mathrm{Recog}(\cdot)$ identifies entities and their relationships, organizing them into triples (V, r, E), where V denotes the head entity, r the relation, and E the tail entity. Relations in the power dispatching knowledge graph are domain-specific and include, but are not limited to:

1. **Management:** (Provincial Dispatch Center, manages, Regional Control Center);
2. **Dependency:** (Forced Energization Operation, depends on, Line Status Confirmation);
3. **Constraint:** (Equipment Operation, constrained by, Voltage Level $\geq$ 220 kV);
4. **Reference:** (Safety Code Article 4.2, refers to, Safety Code Article 3.1);
5. **Applicability:** (Protective Relay, applicable to, 500 kV Substation);
6. **Event Trigger:** (Trip Event, triggers, Maintenance Procedure).

Following extraction, the profiling function $\mathrm{Prof}(\cdot)$ attaches to each entity and relation a normalized name, domain-specific labels (e.g., voltage level, dispatch hierarchy, equipment type, operational state), and a concise description. These metadata support precise matching during retrieval and facilitate semantic linking across documents.

Finally, the deduplication function $\mathrm{Dedup}(\cdot)$ merges duplicate entities or relations appearing in multiple windows, preserving all source references to maintain completeness while reducing storage and retrieval overhead. The process is formally defined as:

$$D = \mathrm{Dedup}(\mathrm{Prof}(V, E)),\ (V, E) \in \mathrm{Recog}(D) \tag{4}$$

where D denotes the final knowledge graph, $\mathrm{Recog}(\cdot)$ is the LLM-based entity–relation extraction function, $\mathrm{Prof}(\cdot)$ is the key–value generation function, and $\mathrm{Dedup}(\cdot)$ is the deduplication function.

Because the entity–relationship extraction step is performed by an LLM, the reliability of the constructed knowledge graph may be affected by extraction errors. Specifically, false negatives may omit important entities or relations, thereby breaking cross-chunk dependency chains and reducing evidence completeness, whereas false positives may introduce spurious nodes or edges that increase retrieval noise. HG-RAG partially mitigates these risks in several ways. First, the overlapping sliding-window strategy preserves local context near chunk boundaries, reducing extraction errors caused by hard truncation. Second, the normalization and deduplication functions help reduce fragmentation caused by inconsistent surface forms across different windows. Third, the knowledge graph in

HG-RAG serves as a structured retrieval scaffold rather than the sole source of evidence; downstream retrieval and answer generation remain grounded in the original text chunks through regex-enhanced retrieval, provenance-aware filtering, and semantic distillation. Therefore, although extraction errors may affect the completeness or precision of the constructed graph, their impact on final QA performance is moderated by the text-grounded retrieval pipeline. Nevertheless, improving graph reliability through confidence-aware extraction and relation verification remains an important direction for future work.

To statistically validate the effectiveness of the proposed sliding-window strategy, we conducted comparative entity extraction experiments on 10 representative power system regulation documents, including dispatch protocols, safety codes, and equipment maintenance guidelines. Under the standard fixed-chunk setting, an average of 245.5 entities (SD = 32.4) were extracted per document. In contrast, the sliding-window strategy in HG-RAG increased the average number of extracted entities to 358.2 (SD = 41.6), corresponding to an average relative improvement of 45.9% in entity coverage. To assess the significance of this improvement, we performed a paired t-test on the extraction results obtained from the same document set under the two segmentation strategies, yielding $p < 0.001$. These results demonstrate that the proposed sliding-window strategy significantly mitigates boundary-induced information loss, improves the completeness of the constructed power dispatching knowledge graph, and provides a stronger foundation for subsequent structured retrieval and reasoning.

3.2. Structured Knowledge Graph Indexing and Storage

Following cross-chunk entity–relationship extraction and the construction of the global knowledge graph, the second stage of HG-RAG is to establish a structured, multi-granularity indexing system. This stage is critical for achieving high retrieval precision, maintaining traceability to source documents, and ensuring scalability in real-world power system environments where document updates are frequent and queries are highly heterogeneous.

3.2.1. Multi-Granularity Index Design

The multi-granularity index in HG-RAG supports both precise factual lookups and complex reasoning tasks by embedding domain-specific labels and association rules into three complementary index structures. Power system queries range from highly specific factual lookups to complex reasoning tasks that require cross-clause dependency resolution. To support this diversity, HG-RAG constructs three complementary indices on top of the knowledge graph. Each index embeds domain-specific business labels (e.g., voltage level, dispatch hierarchy, equipment category, operational state) and association rules (e.g., permission constraints, operational dependency chains, cross-clause references) directly into its structure:

1. **Entity-level index:** Keyed by entity nodes, storing the entity name, aliases, normalized identifiers, and encoded business labels. This design enables constant-time retrieval for exact-match queries under specific operational constraints. For example, Main Transformer–220 kV–Provincial Dispatch–Maintenance serves as a unique key that allows direct matching without vector search overhead.
2. **Relation-level index:** Keyed by relation edges, storing predicate type, associated entity pairs, and applicable association rules. For instance, (Forced Energization Operation, depends on, Line Status Confirmation) can be matched directly, avoiding the high computational cost of path reasoning during answer generation.
3. **Context-level index:** Keyed by the contextual description of entities or relations, with business labels and association rules jointly vectorized to support semantic similarity

search. This index is particularly important for handling fuzzy or implicit queries. For example, "bus coupling" can be mapped to "bus tie switch" and the corresponding operational procedures, even when non-standard terminology is used in the query.

By combining these indices, HG-RAG can dynamically select retrieval strategies that balance recall and precision. Factual queries are served directly via the entity-level index, whereas cross-document reasoning leverages relation- and context-level indices to support multi-hop retrieval.

3.2.2. Scalable Storage Structure

The index data are organized in a key–value (KV) storage model, where the key is a unique identifier derived from entity or relation IDs combined with encoded business labels, and the value stores aggregated descriptions, source metadata, version information, and associated rule lists. The KV abstraction allows the indexing layer to be backend-agnostic, supporting multiple implementations (e.g., JSON document stores, graph databases, distributed KV stores) without altering the retrieval algorithms.

In our current deployment, a lightweight JSON-based storage was used due to its flexibility in schema definition, ease of serialization, and compatibility with both vector databases and full-text search engines. This choice minimizes engineering overhead while supporting rapid integration with heterogeneous retrieval backends.

To handle dynamic knowledge updates, HG-RAG applies the same graph construction pipeline (entity–relation extraction, LLM profiling, deduplication) to newly added or modified document segments, generating a subgraph (V', E'). The update is performed by merging this subgraph into the existing graph (V, E) as:

$$V \leftarrow V \cup V', \; E \leftarrow E \cup E' \tag{5}$$

Only the affected KV index entries are replaced or added, avoiding full index reconstruction. This localized update strategy preserves the global graph structure and index layout while keeping update latency on the order of seconds, which is essential for operational environments where regulations and procedures are frequently revised.

3.3. Suitability

The enhanced RAG module in HG-RAG is explicitly designed for adaptability and scalability across heterogeneous document types and task scenarios. First, both the cross-chunk entity–relationship extraction mechanism and the structured knowledge graph indexing process are grounded in a generic text-to-graph mapping and index-construction workflow. This design does not depend on any specific document format or domain terminology, enabling seamless adaptation to diverse power system document types, including regulations, technical manuals, and operational records. Second, the index storage architecture is decoupled from its implementation technology, allowing deployment flexibility across lightweight JSON stores, graph databases, or distributed KV systems, depending on scale and performance requirements.

Furthermore, the built-in version control and incremental update mechanisms allow the module to operate efficiently in environments where documents are frequently revised, avoiding the computational overhead of full re-indexing. This capability is particularly critical in the power industry, where regulatory standards evolve dynamically and compliance requirements are stringent. Overall, the module not only demonstrates high precision and efficiency on static datasets but also maintains long-term operational stability in real-world production environments, providing the subsequent LLM reasoning stage with a structurally complete, traceable, and continuously updated knowledge base.

4. LLM Reasoning Module in HG-RAG

After the enhanced RAG module completes high-precision candidate retrieval, the LLM reasoning module in HG-RAG integrates the user query with multi-source retrieval results to generate answers that satisfy the stringent accuracy, traceability, and completeness requirements of the power systems domain. This module must not only interpret and disambiguate user intent, but also balance the inclusion of critical evidence with strict token budget constraints. To achieve this, we designed a hierarchical structured reasoning mechanism that combines query parsing, information fusion, and redundancy suppression. This mechanism enables the generation stage to fully exploit structured knowledge and semantic context while maintaining robustness in complex, cross-clause QA tasks.

4.1. Regex-Enhanced Retrieval Module

The regex-enhanced retrieval module ensures recall robustness when knowledge graph coverage is incomplete, the user query is highly specific, or it is phrased in an atypical manner. Although LLM-based entity and relation matching can efficiently retrieve relevant content in most scenarios, its accuracy degrades under these conditions. In high-precision domains such as power system QA—particularly for tasks requiring unique and standardized answers—omitting critical information is unacceptable.

To address these cases, HG-RAG incorporates a regex-enhanced retrieval module that bypasses the knowledge graph and directly matches user queries against all original text chunks. This approach prioritizes exact or near-exact string matches, which is particularly authoritative in the power system domain where, for example, equipment identifiers and operational parameters are expressed in standardized and unique wording that admits no variation, and event timestamps together with schedule codes, such as those describing maintenance windows, outage events, and plan IDs, are recorded in strict and immutable formats. For every authoritative match, the module records its corresponding file_path, providing deterministic provenance information. These file_path values are subsequently used to constrain all downstream retrieval stages to the same source document or documents, thereby eliminating cross-document interference and ensuring that the final answer is grounded in the original, authoritative context. The workflow consists of three stages.

1. Character-level matching

For each text chunk, two character-level metrics are computed:

(1) Normal Match Rate (NMR): the ratio of unique query characters found in the chunk to the total number of unique characters in the query.

(2) Longest Match Rate (LMR): the ratio of the length of the longest consecutive substring in the chunk matching the query to the total query length.

2. Word-level matching

The query and each text chunk are tokenized into words. Both NMR and LMR are then computed at the token level. This step increases robustness to morphological variations, synonyms, and non-standard spacing in the query.

3. Composite scoring and filtering

For each text chunk i, a final composite score s_i is computed as:

$$s_i = \alpha \cdot \mathrm{NMR}_i + \beta \cdot \mathrm{LMR}_i,\ \alpha + \beta = 1 \tag{6}$$

where NMR_i and LMR_i are the normal match rate and longest match rate for chunk i, respectively, and α and β are weighting coefficients satisfying $\alpha + \beta = 1$. Chunks with $s \geq \tau$ are retained as candidates for the subsequent answer generation stage, where τ is a predefined threshold.

This module prioritizes chunks containing long, consecutive query segments, making it particularly effective for technical specifications and regulatory clauses in which wording and formatting are strictly standardized. By operating independently of the knowledge graph, the regex-enhanced retrieval module substantially improves HG-RAG's recall capability and resilience under complex, boundary, and atypical queries, ensuring that no critical content is omitted. The overall workflow is summarized in Algorithm 1.

Regex-based retrieval serves as an effective complement to LLM-based retrieval, particularly for queries with unique and precise requirements. It ensures that even if relevant chunks are missed by the knowledge graph or the LLM, direct text matching can recover them, thereby significantly enhancing recall and robustness in practical applications.

4.2. Hierarchical Structured Retrieval and Distillation

The hierarchical structured retrieval and distillation module in HG-RAG addresses multi-level information needs by combining precise entity matching with broader semantic recall. In structured long-document QA, user queries often contain both fine-grained factual requirements and broader contextual needs. For example, in a power system regulation, a query may request the exact identifier of a specific device (fine-grained) while also requiring its operational role and system context (macro-level). Conventional RAG systems typically rely on a single-layer keyword or vector retrieval strategy, which struggles to meet both requirements simultaneously. Such approaches frequently produce incomplete recall (omitting critical information) or excessive redundancy (introducing irrelevant context).

To address these limitations, HG-RAG employs a hierarchical structured retrieval module in the reasoning stage. This module is designed to maximize both recall and precision under limited token budgets. Given a user query q and a dynamic knowledge graph K, the retrieval operator outputs a candidate set C as:

$$C = \text{distill}(N_{\text{multi}}(K_{\text{core}}(q, K), K_{\text{ext}}(q, K))) \tag{7}$$

where K_{core} denotes core-layer retrieval based on explicit entity or parameter matching, K_{ext} denotes extended-layer retrieval based on implicit thematic recall, N_{multi} performs multi-hop expansion and result merging to capture cross-clause dependencies, where the expansion depth is bounded to two hops in order to balance dependency coverage and retrieval precision, and distill applies semantic distillation to remove redundancy and irrelevance. The retained top-k value is set to $k = 3$, and the matching score uses $w_1 = 0.4$ and $w_2 = 0.6$ for the character-level and word-level components, respectively.

From the perspective of end-to-end information flow, the Regex-Enhanced Retrieval module is first applied to the original text chunks to identify authoritative matches and their associated provenance information, such as 'file_path'. These matches serve as source-consistent anchors for the subsequent hierarchical retrieval process rather than as the final evidence set by themselves. Guided by these anchors, the Hierarchical Structured Retrieval module performs core-layer retrieval for explicit entity or parameter matching, extended-layer retrieval for implicit thematic recall, and bounded two-hop expansion over the knowledge graph to recover cross-clause dependencies. The merged candidate set is then passed to the Semantic Distillation module, which removes redundant or weakly relevant chunks and outputs a compact evidence subset for final answer generation. In this way, Regex-Enhanced Retrieval, Hierarchical Structured Retrieval, and Semantic Distillation form a sequential and complementary pipeline.

Algorithm 1. Regex-Enhanced Retrieval Module

Input: Original text chunks $\mathcal{C} = \{c_1, \ldots, c_N\}$; user query q; match score threshold τ_{regex} (e.g., 0.4); top-K for final selection
Output: Top-K matched chunks for query q
Step 1: Initialization
    Initialize candidate set $\mathcal{C}_{\text{cand}} \leftarrow \varnothing$
Step 2: Regex-based Matching for Each Query
    Extract unique characters and words from query q
for each chunk $c_i \in \mathcal{C}$ **do**
        Character-level matching:
            Compute $\text{NMR}_i^{(\text{char})}$
            Compute $\text{LMR}_i^{(\text{char})}$
        Word-level matching:
            Segment q into words
            Compute $\text{NMR}_i^{(\text{word})}$ and $\text{LMR}_i^{(\text{word})}$
        Score calculation:
            $s_i \leftarrow \alpha \cdot \text{NMR}_i + \beta \cdot \text{LMR}_i,\ \alpha + \beta = 1$
    if $s_i \geq \tau_{\text{regex}}$ **then**
            Add c_i to $\mathcal{C}_{\text{cand}}$ with score s_i
    end if
end for
Step 3: Candidate Selection
    Sort $\mathcal{C}_{\text{cand}}$ by s_i in descending order
    Select top-K chunks as final retrieval results
return Top-K matched chunks for query q

4.2.1. Core-Layer Retrieval and Extended-Layer Retrieval

The core layer focuses on key entities, parameters, or technical terms explicitly mentioned in the query. Named Entity Recognition (NER) and domain terminology matching extract a core keyword set K_{core} from q, followed by exact matching in the knowledge graph K.

$$K_{\text{core}} = \{c \mid \text{Entity}(c) \in K_{\text{core}} \text{or Relation}(c) \in K_{\text{core}}\} \tag{8}$$

This structured matching ensures direct relevance, making it suitable for factual queries. The extended layer targets implicit themes, global background, or contextual cues. Dependency parsing and topic modeling extract an extended keyword set K_{ext} from q, which is used for semantic similarity search in a vector database:

$$K_{\text{ext}} = \{c \mid cos(\nu_{\text{chunk}}, \nu_{\text{ext}}) \geq \theta_{\text{ext}}\} \tag{9}$$

where ν_{chunk} is the vector representation of a candidate chunk, ν_{ext} is the vector of the extended query, and θ_{ext} is the similarity threshold. This process captures semantically relevant content not explicitly linked in the KG, making it suitable for macro-level or integrative questions.

4.2.2. Multi-Hop Expansion and Result Merging

To enhance cross-clause reasoning while avoiding excessive graph traversal, the module performs bounded multi-hop neighbor expansion over the seed entities retrieved from the core layer and the extended layer. Let H denote the maximum hop number. Starting

from the seed entity set, HG-RAG recursively expands graph neighbors up to H hops and collects the reachable nodes and their associated text blocks as candidate evidence:

$$N_{\text{multi}} = \{c' \mid v \in K_{\text{core}} \cup K_{\text{ext}}, (v, c') \in E\} \tag{10}$$

In this work, the maximum hop number was set to $H = 2$. This setting was chosen according to the characteristics of power-system documents, where most useful dependency chains are relatively local and can typically be captured within one or two relational steps. Expanding beyond two hops is more likely to introduce weakly relevant or redundant context, thereby increasing token consumption and reducing retrieval precision.

The merged candidate set after this step is:

$$C_{\text{candidates}} = K_{\text{core}} \cup K_{\text{ext}} \cup N_{\text{multi}} \tag{11}$$

If the regex-enhanced retrieval stage has produced authoritative matches, the file_path values associated with those matches are used at this point to filter the candidate set, retaining only chunks originating from the same source documents. This restriction is applied to prevent interference from other documents that may contain superficially similar but contextually different information, which could otherwise lead to incorrect answers. The provenance-based filtering is skipped when no regex matches are available, ensuring that recall is not unnecessarily reduced.

4.2.3. Semantic Distillation

All candidate text chunks from the filtered set undergo semantic consistency filtering to retain only the most relevant information, significantly reducing noise and redundancy.

Let the retrieved candidate set be $C_{\text{candidates}}$, with each chunk c_i having description text t_i and semantic vector e_i encoded by a pretrained model (e.g., BAAI/bge-large-zh-v1.5). We first compute the mean vector:

$$\bar{e} = \frac{1}{N} \sum_{i=1}^{N} e_i \tag{12}$$

Then, for each chunk c_i, we compute cosine similarity with the mean vector:

$$\text{sim}_i = cos(e_i, \bar{e}) \tag{13}$$

Finally, we retain only those chunks meeting the similarity threshold τ (e.g., 0.75):

$$C_{\text{filtered}} = \{c_i \mid \text{sim}_i \geq \tau\} \tag{14}$$

Through hierarchical structured retrieval and semantic distillation, HG-RAG satisfies users' multi-level information needs for both critical details and global context, significantly improving the comprehensiveness, relevance, and usability of answers. Experimental results and case analyses show that this mechanism outperforms conventional single-layer retrieval methods in power system document QA, producing high-quality answers with both depth and breadth.

4.3. Adaptability

The LLM reasoning module in HG-RAG demonstrates strong adaptability, enabling it to handle diverse types of power system documents and varied user query requirements with flexibility. First, both the core-extended query decomposition mechanism and the regex-enhanced retrieval module are designed in a domain-agnostic manner, independent of specific document structures or terminologies, allowing seamless adaptation to regu-

lations, technical manuals, maintenance records, and other document types. Second, the semantic distillation module applies semantic consistency filtering to effectively address challenges in long documents, complex contexts, and multi-turn QA scenarios, significantly reducing the interference of redundant information on LLM reasoning and improving efficiency and stability under real-world deployment conditions such as high concurrency and low latency. Furthermore, HG-RAG supports incremental knowledge base updates and multi-source integration, enabling continuous incorporation of new knowledge and adaptation to the evolving standards and operational workflows of the power industry. In summary, the LLM reasoning module in HG-RAG not only achieves high accuracy and robustness but also offers excellent scalability and engineering deployability, providing a solid foundation for intelligent QA and knowledge management in power systems.

5. Case Study

5.1. Dataset Construction

To rigorously evaluate the performance of different RAG frameworks in the power systems domain under strict confidentiality constraints, we constructed an LLM-assisted QA benchmark from real and heterogeneous documents from actual power grid operations. Specifically, the source corpus consists of authentic power-dispatch documents, while the question–answer pairs were generated from these documents by a local large language model and then manually reviewed by domain experts. Specifically, it consisted of 60 real power-dispatch documents for short-answer questions and 30 documents for multiple-choice questions. After preprocessing, the overall corpus contained approximately 200,000 tokens. All documents were processed locally, and no raw source text was exposed to external systems. A local large language model (Qwen3-14B) was then used to generate question–answer pairs, followed by manual expert review. During this review process, each QA pair was examined according to the following criteria: (1) factual correctness with respect to the source document; (2) linguistic clarity and completeness of the question and answer; (3) consistency with power-system terminology and dispatching practice; and (4) alignment with the intended question category, namely conceptual understanding, factual details, or analytical reasoning. For multiple-choice questions, the review additionally verified that each item had one unambiguous correct answer and plausible distractors. QA pairs that did not satisfy these criteria were revised or discarded.

5.1.1. Short-Answer QA Set

The source corpus was segmented into 93 chunks, each containing approximately 1200 tokens. A total of 1500 short-answer questions were generated, evenly distributed across three categories, each containing 500 questions:

1. **Conceptual Understanding:** Questions that require summarizing core concepts or describing overall operational principles stated in the document.
2. **Factual Details:** Questions that require retrieving specific facts, numerical values, or clause-level details without inference.
3. **Analytical Reasoning:** Questions that require integrating multiple clauses or entities to derive an answer through reasoning or application.

Representative examples of each category are provided in Table 1.

Table 1. Short-answer question categories.

Category	Question	Answer
Conceptual Understanding	"What aspects of power grid operation does this document primarily address?"	System frequency recovery, system oscillation, system stability failure, secondary equipment anomalies
Factual Details	"How many minutes after a line trip should the first forced energization be performed?"	20
Analytical Reasoning	"What is the label of the first grounding disconnector group ('JIA') for busbar 1M?"	221JIA00

5.1.2. Multiple-Choice QA Set

The source corpus for multiple-choice questions was segmented into 26 chunks, each containing approximately 1200 tokens. A total of 1500 multiple-choice questions were generated, also evenly distributed across the same three categories to ensure comparability with the short-answer set:

1. **Conceptual Understanding:** Requires identifying conceptual definitions or high-level descriptions from the document.
2. **Factual Details:** Requires recognizing explicit facts, parameter values, or regulation statements.
3. **Analytical Reasoning:** Requires applying document content to scenario-based reasoning or decision-making.

Representative examples of each category are provided in Table 2.

Table 2. Multiple-choice question categories.

Category	Question	Options & Answer
Conceptual Understanding	"Which of the following is a responsibility of the provincial dispatch center?"	A. Equipment procurement B. Unified dispatch C. Line design D. Asset evaluation
Factual Details	"What is the rated voltage of the main transformer in the substation?"	A. 110 kV B. 220 kV C. 330 kV D. 500 kV
Analytical Reasoning	"If a 110 kV line supplying both a hospital and a factory experiences a fault, after fault isolation, which action should operators take first?"	A. Restore factory power first B. Immediately replace faulty equipment C. Conduct a grid inspection D. Restore hospital power first

5.2. Experimental Settings

To ensure reproducibility and transparency, all key configurations for dataset generation, retrieval, and answer verification are summarized in Table 3. Specifically, this table details the primary parameters used in the HG-RAG pipeline, the local LLM models employed for QA dataset generation and answer verification, as well as the embedding

models and other critical thresholds. Unless otherwise specified in the corresponding figure or table captions, all experiments were conducted under these unified settings.

Table 3. Configuration settings for evaluations.

Parameter	Value	Description
RAG Configuration		
Chunk size (tokens)	1200	Tokens per chunk
Sliding step size (tokens)	1000	Shift between adjacent windows
Top-k chunks	3	Number of matched chunks
PDF Embedding Model	BAAI/bge-m3	For initial document chunking
Chunk Cleaning Model	BAAI/bge-large-zh-v1.5	For semantic similarity filtering
LLM Configuration		
QA Generation Model	Qwen3-14B	For Q&A dataset generation
Answering Model	Qwen3-8B	For answer validation
Thought Mode	Enabled	LLM reasoning mode
Other Settings		
NMR Weight	0.4	For NMR
LMR Weight	0.6	For LMR
Regex Match Threshold	0.4	For retrieval
Semantic Similarity Threshold	0.75	For chunk cleaning

5.3. Evaluation Metrics and Results

To evaluate HG-RAG against LightRAG [27] and GraphRAG [28] on the constructed power system QA dataset, we used two practical metrics:

(1) **Accuracy:** An answer is considered correct if the LLM output contains the ground-truth answer as a substring, regardless of additional context or minor wording variations. This lenient matching reflects real operational practice, where domain experts primarily care that the key factual element is present.

(2) **Token Consumption:** Reported as the total number of tokens used for both retrieval and generation per question, as this aggregate cost directly affects deployment feasibility.

All methods were evaluated under identical experimental conditions using the same dataset and document collection to ensure comparability. Specifically, LightRAG and GraphRAG were implemented using their default or officially recommended hyperparameter settings, and no additional task-specific hyperparameter tuning was performed for these baseline methods. HG-RAG uses the fixed parameter configuration reported in Table 3.

5.3.1. Short-Answer Question Results

Figure 3 shows the performance of the three RAG frameworks on short-answer questions. HG-RAG achieved an overall accuracy of 85.47%, significantly outperforming LightRAG (12.30%) and GraphRAG (21.73%), and maintained a lead across all three task categories. The gains in Factual Details (91.4%) and Conceptual Understanding (86.7%) primarily stemmed from the core-layer retrieval and semantic distillation mechanisms, which precisely match entities and filter redundancy. The improvement in Reasoning/Integrative tasks (78.4%) was mainly attributed to extended retrieval and multi-hop expansion, which complete cross-clause dependencies. Average token consumption was only 11,330.67, substantially lower than the competing methods, demonstrating the unification of high precision and high efficiency.

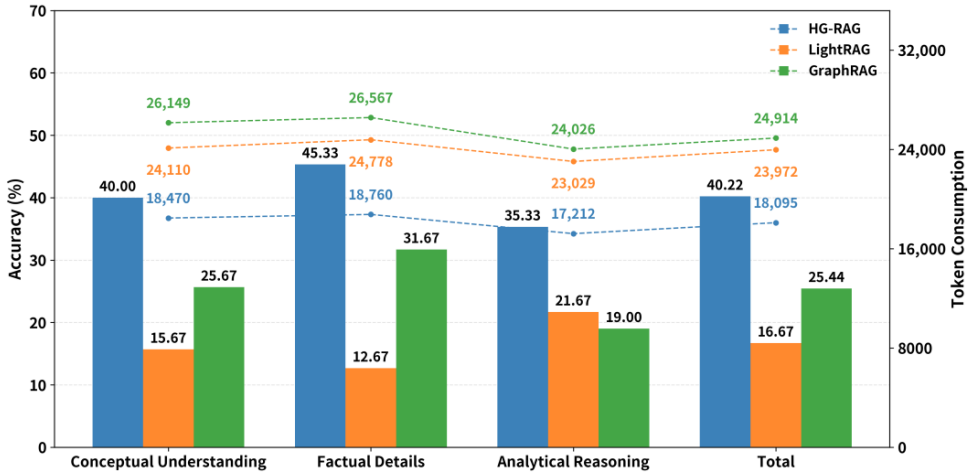


Figure 3. Performance of HG-RAG, LightRAG, and GraphRAG on short-answer questions.

5.3.2. Multiple-Choice Question Results

Figure 4 presents the results for multiple-choice questions. Compared with short-answer tasks, all methods showed a decrease in overall accuracy, with HG-RAG reaching 40.22%, still significantly higher than LightRAG (16.67%) and GraphRAG (25.44%). By task type, HG-RAG's advantage waws most pronounced in Fact Recognition (45.33%) and Clause Understanding (40.00%), mainly due to the regex-enhanced retrieval mechanism, which recovers critical information missed by the knowledge graph. The overall accuracy drop relative to short-answer questions can be attributed to several factors. First, the source documents for multiple-choice tasks are generally longer and contain denser cross-clause references, which increases retrieval difficulty and introduces more semantically similar but irrelevant context. Second, the distractor options are usually plausible rather than arbitrary, and often share similar entities, terminology, or numerical patterns with the correct answer. Therefore, retrieving partially relevant evidence may still be insufficient for correct option selection. Third, unlike short-answer questions, where the response is considered correct as long as the key answer element is present, multiple-choice questions require an additional decision step in which the model must compare all candidate options and identify the single best one. This places higher demands on both fine-grained evidence alignment and final decision precision. Finally, some questions require not only locating the relevant clauses but also correctly mapping the retrieved evidence to the corresponding option, which further increases the risk of misclassification.

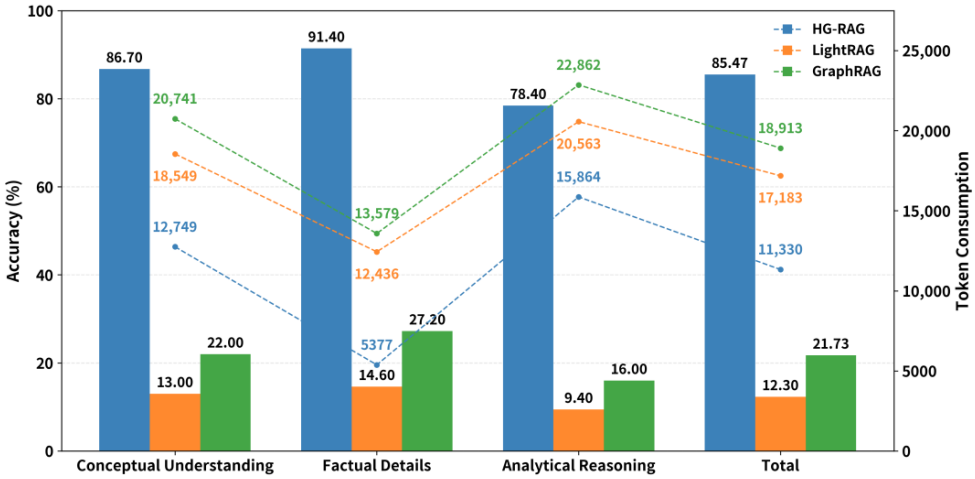


Figure 4. Performance of HG-RAG, LightRAG, and GraphRAG on multiple-choice questions.

In terms of token consumption, HG-RAG averaged 18,095.96 tokens per multiple-choice question—higher than in short-answer tasks (11,330.67), but still substantially lower than LightRAG (23,972.79) and GraphRAG (24,914.60) across all task categories. This difference was mainly due to the structural characteristics of the source documents: multiple-choice tasks are associated with longer documents and denser cross-clause references, requiring the retrieval and processing of more context to ensure option discrimination accuracy.

Interestingly, within multiple-choice tasks, Reasoning/Integrative questions consume fewer tokens (17,212.64) than Conceptual Understanding (18,470.03) and Fact Recognition (18,760.56)—the opposite trend from short-answer tasks. This is because reasoning-type multiple-choice questions, despite involving cross-clause dependencies, have a constrained answer space defined by the provided options. In the generation stage, the system does not need to narrate the full reasoning chain but only retain key evidence chunks directly relevant to option discrimination. Structured retrieval and semantic distillation quickly prune irrelevant context, reducing the number of tokens passed to the LLM.

This contrast shows that HG-RAG dynamically adjusts the scope of retrieved content based on task type and document structure: for open-ended short-answer reasoning tasks, it retains a complete evidence chain to ensure coverage, while for closed multiple-choice reasoning tasks, it prioritizes context compression to lower computational cost while preserving essential decision-making information.

5.3.3. Incremental Knowledge Retention Experiment

To further evaluate the stability and scalability of HG-RAG in dynamic knowledge update scenarios, we designed an incremental knowledge retention experiment focusing on the system's ability to preserve existing knowledge after adding new documents. Since GraphRAG does not support incremental updates, we compared HG-RAG with LightRAG, which is capable of handling dynamic document additions.

The experiment first builds an initial knowledge graph from the short-answer question documents. Subsequently, another real-world power system document corresponding to the multiple-choice questions is incrementally added. The reasoning module is then re-executed on the same short-answer test set, and performance changes are measured. Table 4 summarizes the results.

Table 4. Knowledge retention under incremental updates.

Scenario	Accuracy (%)	Tokens
HG-RAG (Short-answer only)	85.47	11,330.67
HG-RAG (Short + MC docs)	74.92	11,350.12
HG-RAG *w/o* file_path constraint (Short + MC docs)	58.35	15,640.28
LightRAG (Short-answer only)	12.30	17,183.07
LightRAG (Short + MC docs)	6.50	18,340.55

Results show that after incremental updates, HG-RAG's short-answer accuracy decreased from 85.47% to 74.92%, corresponding to a relative drop of 12.35%, whereas LightRAG suffered a much larger relative drop from 12.30% to 6.50%. Token usage for HG-RAG remained nearly unchanged, while LightRAG's token consumption increased significantly.

The robustness of HG-RAG in this scenario is largely due to its Regex-Enhanced Retrieval module. To quantitatively evaluate the effect of the 'file_path' constraint, we further conducted an ablation test by disabling this source-consistency filtering mechanism under the same incremental setting (Short + MC docs). As shown in Table 4, without the 'file_path' constraint, the accuracy of HG-RAG decreased from 74.92% to 58.35%, while

token usage increased from 11,350.12 to 15,640.28. This corresponds to an accuracy gain of 16.57 percentage points brought by the proposed constraint.

In our dataset, the short-answer and multiple-choice documents shared highly similar content, leading to overlapping entities and relations. Without source-aware filtering, retrieval may return semantically plausible but document-inconsistent entities, thereby introducing cross-document noise and reducing answer accuracy. HG-RAG addresses this issue by first using authoritative matches from the Regex-Enhanced Retrieval module to identify the correct 'file_path' values, and then restricting subsequent hierarchical retrieval and reasoning to chunks from these source documents. These quantitative results confirm that the 'file_path'-constrained retrieval strategy substantially mitigates cross-document interference when highly similar documents coexist in the knowledge base.

In addition, HG-RAG's version control and deduplication-merging mechanisms ensure that newly added documents do not overwrite or obscure knowledge from existing ones, thereby preserving completeness and consistency. This design enables high-accuracy, low-latency updates in long-term online maintenance scenarios—an essential capability in the power system domain where regulations and operational procedures are frequently revised.

6. Ablation Study

To further assess the contribution of each key module in HG-RAG to overall performance, we conducted an ablation study on the short-answer tasks, focusing on the impact of the semantic distillation module and the regex-enhanced retrieval module on system accuracy and efficiency. The results are summarized in Table 5.

Table 5. Ablation study results on short-answer QA tasks.

Method	Semantic Distillation	Regex Recall	Accuracy (%)	Tokens
HG-RAG	✓	✓	85.47	11,330.67
	×	✓	60.25	21,579.54
	✓	×	54.83	11,784.56
	×	×	22.74	26,685.41
LightRAG	×	×	12.30	17,183.07
	×	✓	57.60	18,749.63
	✓	×	42.40	13,369.42
	✓	✓	66.20	11,750.67
GraphRAG	×	×	21.73	18,913.83
	×	✓	67.98	20,425.77
	✓	×	58.67	14,416.52
	✓	✓	75.22	24,134.87

Removing semantic distillation from HG-RAG caused accuracy to drop sharply from 85.47% to 60.25% and nearly doubled token usage from 11,330.67 to 21,579.54, confirming that this module plays a critical role in removing redundant chunks, compressing input length, and improving LLM focus. In contrast, removing regex-enhanced retrieval reduced accuracy to 54.83% with minimal change in token usage, indicating that its primary benefit lies in recall robustness, particularly when knowledge coverage is incomplete or the query expression varies. When both modules were disabled, accuracy fell dramatically to 22.74% and token consumption rose to 26,685.41, highlighting the severity of performance degradation without these components.

The two modules also demonstrated strong cross-framework generalizability. Integrating both into LightRAG increased accuracy from 12.30% to 66.20% while reducing token

usage from 17,183.07 to 11,750.67. For GraphRAG, the same integration raised the accuracy from 21.73% to 75.22%, although token usage increased due to the framework's broader retrieval scope interacting with regex matches. These results confirm that the modules are not specific to HG-RAG and can substantially enhance other RAG architectures.

Rather than contributing independently, semantic distillation and regex-enhanced retrieval interact in a sequential and complementary manner within HG-RAG. Regex-enhanced retrieval first improves authoritative evidence recall by directly matching structured query patterns in the source text, such as equipment identifiers, clause references, and other domain-specific expressions. However, the retrieved candidate set may still contain redundant or partially relevant chunks. Semantic distillation is then applied to compress and refine these retrieved results before generation, thereby improving prompt quality and reducing token consumption.

This interaction is also reflected quantitatively in Table 5. When semantic distillation is enabled, adding regex-enhanced retrieval improved accuracy from 54.83% to 85.47% while keeping token usage nearly unchanged (11,784.56 vs. 11,330.67). Conversely, when regex-enhanced retrieval was enabled, adding semantic distillation improved the accuracy from 60.25% to 85.47% and reduced the token usage from 21,579.54 to 11,330.67. When both modules were disabled, accuracy dropped sharply to 22.74%, indicating that neither module alone is sufficient. These results show that regex-enhanced retrieval mainly provides high-recall source grounding, while semantic distillation converts the expanded evidence set into compact and high-precision context. Their combination therefore enables HG-RAG to achieve both high accuracy and low token cost.

7. Discussion

Although HG-RAG delivers substantial improvements in accuracy, efficiency, and traceability, several technical directions could further enhance its applicability in real-world power system operations.

First, with respect to explainability and process transparency, the current system returns matched text fragments together with their source locations. This design provides point-level traceability, but it does not yet fully expose the complete process of retrieval and evidence aggregation. Because HG-RAG already organizes evidence through hierarchical relations among entities, chunks, and source documents, a practical extension is to introduce path-level provenance tracking that explicitly records the sequence from the initial regex/entity match, through hierarchical retrieval and multi-hop expansion, to the final supporting evidence. The activated nodes and relations could then be presented as auditable subgraphs, while each reasoning path could be assigned an importance score or a confidence score for compliance inspection. In addition, answer generation could be constrained to evidence-grounded output, such that each conclusion is explicitly aligned with the cited source chunks.

Second, multimodal fusion remains limited. Important information in power regulations and operational records is often embedded in tables, diagrams, equipment schematics, and nameplates, which cannot be fully utilized by the current text-centric framework. A concrete extension is to construct a multimodal knowledge graph by combining OCR and layout parsing for document images, vision-language encoders for diagrams and schematics, and cross-modal links between textual entities and visual or tabular regions. On top of this graph, modality-aware retrieval and reranking could be introduced so that textual, tabular, and visual evidence are considered jointly during answer generation.

Third, long-chain reasoning over numerous entities and clauses may introduce noise or omit critical intermediate evidence. To address this issue, future versions of HG-RAG could incorporate query-conditioned path pruning to remove low-relevance nodes during

multi-hop expansion, hop-wise verification and reranking to retain only evidence chains supported by source text, and reasoning-chain compression to transform long retrieved passages into compact structured evidence units before final generation. These mechanisms could reduce the accumulation of irrelevant context, control the propagation of errors, and improve the robustness of reasoning in long-document settings.

8. Conclusions

This work addresses the need for high-accuracy, high-efficiency, and traceable QA over structured long documents in the power systems domain by introducing HG-RAG—a framework that builds a globally consistent knowledge graph via sliding-window entity–relation extraction, combines multi-granularity structured indexing with hierarchical structured retrieval, and integrates two key modules: semantic distillation and regex-enhanced retrieval. This design ensures comprehensive recall while minimizing redundancy, achieving a balanced trade-off between precision, efficiency, and robustness.

Extensive evaluation on real-world power regulation datasets shows that HG-RAG consistently outperforms representative baselines across multiple task types, improving accuracy while reducing average token consumption. The ablation study confirms that semantic distillation is essential for input compression and generation focus, while regex-enhanced retrieval safeguards recall in scenarios with incomplete knowledge coverage and diverse query expressions; their synergy underpins HG-RAG's strong performance.

While the current system already meets many operational requirements, future research will focus on multimodal knowledge integration, enhanced reasoning-chain management, and interactive clarification QA to further improve adaptability in complex and evolving environments. By combining methodological innovation with engineering practicality, HG-RAG offers a viable pathway for intelligent QA and knowledge management in power systems, with strong potential to accelerate the sector's digital transformation and intelligent upgrading.

Author Contributions: Conceptualization, Z.S.; methodology, X.C.; software, Z.H.; validation, B.N.; formal analysis, Z.H.; resources, T.Y.; data curation, Z.M.; writing—original draft preparation, Z.H.; writing—review and editing, T.Y.; visualization, Z.S.; supervision, Z.S.; project administration, X.C.; funding acquisition, Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Program of China Southern Power Grid Co., Ltd., under Grant No. 036000KC23090003 (GDKJXM20231024). This project is a key scientific and technological project of China Southern Power Grid, funded by the Power Dispatching and Control Center of Guangdong Power Grid Co., Ltd.

Data Availability Statement: The dataset used in this study can be made available by the corresponding author upon reasonable request.

Conflicts of Interest: Authors Zhijun Shen, Xinlei Cai, Binye Ni, and Zijie Meng were employed by the Power Dispatch Control Center of Guangdong Power Grid Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Chen, J.; Lu, G.; Pan, Z.; Yu, T.; Ding, M.; Yang, H. Research Review of the Knowledge Graph and Its Application in Power System Dispatching and Operation. *Front. Energy Res.* **2022**, *10*, 896836. [CrossRef]
2. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Yu, P.S. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 494–514. [CrossRef]
3. Zhao, P.; Zhang, H.; Yu, Q.; Wang, Z.; Geng, Y.; Fu, F.; Yang, L.; Zhang, W.; Jiang, J.; Cui, B. Retrieval-Augmented Generation for AI-Generated Content: A Survey. *Data Sci. Eng.* **2026**, *11*, 1–29. [CrossRef]

4. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
5. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599. [CrossRef]
6. Youn, J.; Tagkopoulos, I. KGLM: Integrating Knowledge Graph Structure in Language Models for Link Prediction. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (SEM 2023), Toronto, Canada*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 217–224. [CrossRef]
7. Park, S.; Chen, W.; Han, D.; Tanneau, M.; Van Hentenryck, P. Confidence-Aware Graph Neural Networks for Learning Reliability Assessment Commitments. *IEEE Trans. Power Syst.* **2024**, *39*, 3839–3850. [CrossRef]
8. Bernier, F.; Cao, J.; Cordy, M.; Ghamizi, S. PowerGraph-LLM: Novel Power Grid Graph Embedding and Optimization with Large Language Models. *IEEE Trans. Power Syst.* **2025**, *40*, 5483–5486. [CrossRef]
9. Jia, M.; Cui, Z.; Hug, G. Enhancing LLMs for Power System Simulations: A Feedback-Driven Multi-Agent Framework. *IEEE Trans. Smart Grid* **2025**, *16*, 5556–5572. [CrossRef]
10. Saberi, H.; Zhang, C.; Dong, Z.Y. A Multi-Agent Deep Constrained Q-Learning Method for Smart Building Energy Management Under Uncertainties. *IEEE Trans. Smart Grid* **2024**, *15*, 4649–4661. [CrossRef]
11. Liu, G.; Tang, Y.; Guo, W.; Dang, J. Redesigning Power Grid Dispatching Automation Systems by Integrating LLMs, KGs, and AI Agents. *CSEE J. Power Energy Syst.* **2025**, *11*, 2610–2622. [CrossRef]
12. Wang, Y.; Lipka, N.; Rossi, R.A.; Siu, A.; Zhang, R.; Derr, T. Knowledge graph prompting for multi-document question answering. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 19206–19214. [CrossRef]
13. Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.M.; Shum, H.-Y.; Guo, J. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In Proceedings of the International Conference on Learning Representation, Vienna, Austria, 7–11 May 2024.
14. Si, R.; Chen, S.; Zhang, J.; Xu, J.; Zhang, L. A multi-agent reinforcement learning method for distribution system restoration considering dynamic network reconfiguration. *Appl. Energy* **2024**, *372*, 123625. [CrossRef]
15. Liu, H.; Wang, Z.; Chen, X.; Li, Z.; Xiong, F.; Yu, Q.; Zhang, W. HopRAG: Multi-Hop Reasoning for Logic-Aware Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 1897–1913. [CrossRef]
16. Arslan, M.; Mahdjoubi, L.; Munawar, S. Driving sustainable energy transitions with a multi-source RAG-LLM system. *Energy Build.* **2024**, *324*, 114827. [CrossRef]
17. Ruan, J.; Liang, G.; Zhao, H.; Liu, G.; Sun, X.; Qiu, J.; Xu, Z.; Wen, F.; Dong, Z.Y. Applying Large Language Models to Power Systems: Potential Security Threats. *IEEE Trans. Smart Grid* **2024**, *15*, 3333–3336. [CrossRef]
18. Yao, Y.; Zhang, X.; Wang, J.; Ding, F. Multi-Agent Reinforcement Learning for Distribution System Critical Load Restoration. In *2023 IEEE Power & Energy Society General Meeting (PESGM)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5. [CrossRef]
19. Xu, K.; Zhang, K.; Li, J.; Huang, W.; Wang, Y. CRP-RAG: A Retrieval-Augmented Generation Framework for Supporting Complex Logical Reasoning and Knowledge Planning. *Electronics* **2024**, *14*, 47. [CrossRef]
20. Mao, J.; Zheng, C.; Liu, W.; Zhang, Z. MGRAG: MultiModal grid-aware retrieval augmentation generation framework for power grid work tickets. *Pattern Recognit.* **2026**, *169*, 111845. [CrossRef]
21. Ye, Z.; Qi, D.; Liu, H.; Zhang, S. IHGR-RAG: An Enhanced Retrieval-Augmented Generation Framework for Accurate and Interpretable Power Equipment Condition Assessment. *Electronics* **2025**, *14*, 3284. [CrossRef]
22. Lu, Y.; Peng, J.; Xu, X.; He, Y.; Li, T.; Wei, J.; Jing, H.; Wang, H.; Xu, B.; Song, H. A Retrieval-Augmented Generation Framework for Electric Power Industry Question Answering. In Proceedings of the 2024 2nd International Conference on Electronics, Computers and Communication Technology, Chengdu, China, 25–27 October 2024; pp. 95–100. [CrossRef]
23. Wang, C.; Hua, M.; Song, J.; Tang, X.-s. Knowledge Graphs Enhanced Large Language Model Prompt for Electric Power Question Answering. In Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, 20–22 October 2023; pp. 24–29. [CrossRef]
24. Zhao, J.; Ma, Z.; Zhao, H.; Zhang, X.; Liu, Q.; Zhang, C. Self-consistency, Extract and Rectify: Knowledge Graph Enhance Large Language Model for Electric Power Question Answering. In *Advanced Intelligent Computing Technology and Applications, 20th International Conference, ICIC 2024, Tianjin, China, 2024*; Springer Nature: Singapore, 2024; pp. 493–504. [CrossRef]
25. Papageorgiou, G.; Sarlis, V.; Maragoudakis, M.; Tjortjis, C. A Multimodal Framework Embedding Retrieval-Augmented Generation with MLLMs for Eurobarometer Data. *AI* **2025**, *6*, 50. [CrossRef]
26. Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; Leskovec, J. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, June 2021*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 535–546. [CrossRef]

27. Zhou, B.; Li, K. Fusing Geoscience Large Language Models and Lightweight RAG for Enhanced Geological Question Answering. *Geosciences* **2025**, *15*, 382. [CrossRef]
28. Naganawa, H.; Hirata, E.; Yamada, A. Implementing a Knowledge Management System with GraphRAG: A Physical Internet Example. *Electronics* **2025**, *14*, 4948. [CrossRef]
29. Hu, Z.; Ma, R.; Wang, B.; Huang, Y.; Su, R. A General Resiliency Enhancement Framework for Load Frequency Control of Interconnected Power Systems Considering Internet of Things Faults. *IEEE Trans. Ind. Inform.* **2024**, *20*, 10695–10705. [CrossRef]
30. Hu, Z.; Liu, S.; Luo, W.; Wu, L. Resilient Distributed Fuzzy Load Frequency Regulation for Power Systems Under Cross-Layer Random Denial-of-Service Attacks. *IEEE Trans. Cybern.* **2022**, *52*, 2396–2406. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.