GridCodex: A RAG-Driven AI Framework for Power Grid Code Reasoning and Compliance

Jinquan Shi[1], Yingying Cheng[1], Fan Zhang[1], Miao Jiang[2], Jun Lin[2], Yanbai Shen[2]

[1]Theory Lab, 2012 Labs, Huawei Technologies Co., Ltd.
[2]Digital Energy BU, Huawei Technologies Co., Ltd.
{shi.jinquan, cheng.yingying1, zhang.fan2, jiangmiao8, linjun37, shenyanbai}@huawei.com

Abstract

The global shift towards renewable energy presents unprecedented challenges for the electricity industry, making regulatory reasoning and compliance increasingly vital. Grid codes—the regulations governing grid operations—are complex and often lack automated interpretation solutions, which hinders industry expansion and undermines profitability for electricity companies. We introduce GridCodex, an end-to-end framework for grid code reasoning and compliance that leverages large language models and retrieval-augmented generation (RAG). Our framework advances conventional RAG workflows through multi-stage query refinement and enhanced retrieval with RAPTOR. We validate the effectiveness of GridCodex with comprehensive benchmarks, including automated answer assessment across multiple dimensions and regulatory agencies. Experimental results showcase a 26.4% improvement in answer quality and more than a 10-fold increase in Recall@30. An ablation study further examines the impact of base model selection.

Introduction

In the accelerating transition toward renewable energy, electricity has become the central medium linking diverse sustainable energy sources with end users, placing unprecedented demands on the robustness and efficiency of power grids (Gri 2016; Ren 2017; Ntomaris et al. 2014). Grid codes are a set of rule and regulations that govern the reliable operation of power-generating equipment to the electricity grid. They are established by grid operators to ensure the safety, reliability, and efficiency of the electricity supply, and they cover aspects like voltage and frequency control, power quality, and how equipment must perform during faults.

Grid code reasoning is the process of analyzing and interpreting mandates for power-generating units and users to ensure a power system or component complies with them. Specifically, grid code reasoning can be used to support some practical applications. For example, providers of digital power products and solutions have to make sure that their products comply with the local regulations. Engineers and planners can check the product details against the local grid code for understanding and applying complex regulatory provisions, facilitating faster onboarding and reducing the risk of compliance errors. Another case is annual regulatory compliance verification. Power grid operators can query whether specific operational procedures (e.g., voltage control, frequency response, or renewable integration strategies) comply with national or regional grid codes.

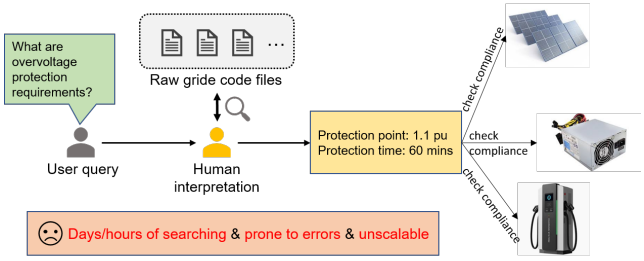

Figure 1: Conventional human grid code interpretation workflow. Specialists manually read and verify raw grid code files to answer user queries, which is time-consuming and prone to errors.

However, grid codes differ significantly across countries and regions (Ullah et al. 2025), rendering grid code compliance a complex but indispensable challenge for electricity infrastructure providers seeking international expansion. Conventional grid code interpretation relies heavily on human expertise, as illustrated in Figure 1. Grid codes are often written with dense, region-specific terminology, technical jargon, and implicit assumptions that demand deep domain knowledge. To navigate these complexities, electricity providers typically consult local specialists or external agencies, which significantly increases operational costs and project timelines. Human interpretation, however, is prone to inconsistency, stemming from ambiguities in the documents or simple human error. The challenge intensifies with lengthy, multi-layered regulations, where manual review becomes prohibitively time-consuming and costly. This dilemma is further compounded for companies operating across multiple jurisdictions, where the limited capacity of specialists creates a critical bottleneck. These challenges underscore the urgent need for scalable, automated solutions that can efficiently process large volumes of regulatory text, deliver reliable compliance interpretation, and reduce the risk of errors.

The rapid advancement of large language models (LLMs)

Method	Domain Adaptability	Deployment Cost	Data Requirement	Long Doc Handling
General LLM	✗ Low	✓ Low	✗ None (pretrained only)	✗ Poor
Fine-Tuning	✓ High	✗ High	✗ Requires labeled data	✗ Moderate
RAG	✓ Medium	✓ Medium	✓ Unlabeled documents	✗ Varies
Improved Retrieval	✓ High	✓ Medium	✓ Unlabeled documents	✓ Strong

Table 1: Comparison of mainstream approaches for industry-specific LLM adaptations.

(OpenAI 2023; Int 2024; Minaee et al. 2025) offers an intriguing solution for grid code reasoning and compliance. With extraordinary text comprehension ability, LLMs excel at text understanding and question answering tasks. However, lack of specialized domain knowledge and susceptibility to hallucinations (Xu, Jain, and Kankanhalli 2025; Huang et al. 2025) challenge their enterprise and industrial applications. Table 1 have summarized mainstream LLM domain adaptation methods. While fine-tuning with domain-specific datasets can significantly mitigate some of these issues (J et al. 2024; Jeong 2024), it requires high-quality datasets, which are difficult to obtain in energy industry due to the data confidentiality and cumbersome work of data cleaning and formatting. Additionally, substantial computation resource and LLM engineering experience required can hinder the fine-tuning of LLMs (Hu et al. 2021; Chen et al. 2024), especially in mid-size or small enterprises and conventional industries, such as energy and power grids.

Recently, retrieval-augmented generation (RAG) has emerged as a cost-effective alternative to fine-tuning, offering higher accuracy and more flexible deployment for enterprise applications (Lewis et al. 2021; Fan et al. 2024). By enabling LLMs to seamlessly access external knowledge bases during inference, RAG reduces hallucinations and improves factual consistency without retraining. Constructing such knowledge bases is straightforward: raw documents are split into chunks, embedded, and stored in vector databases—an approach that can be fully deployed locally without extensive data cleaning. As a result, RAG provides a private, scalable, and low-overhead solution for adapting LLMs to industrial use cases (Balaguer et al. 2024).

In this paper, we introduce a RAG-driven AI framework for grid code interpretation, leveraging domain-specific knowledge bases constructed from regulatory documents and orchestrated with high-performance open-source LLMs such as DeepSeek (Guo et al. 2025; Liu et al. 2024) and Qwen3 (Yang et al. 2025). These models exhibit strong reasoning capability and scalability, enabling reliable compliance interpretation. An LLM-based automated scoring system has been applied to conduct extensive benchmarks on real-world grid codes, which demonstrates that our solution achieves answer quality of up to 87.95 % and faithfulness up to 96.90 %. Beyond compliance interpretation, our framework also holds promise for proactive regulatory monitoring, such as detecting potential violations and generating simulation-ready grid configurations, thereby accelerating regulatory workflows and mitigating compliance risks in the electricity sector.

Our contributions are described as follows:

- **First RAG-driven framework for power grid code reasoning and compliance.** We present the first RAG-based agent tailored for automated reasoning and compliance verification in grid codes, addressing domain-specific challenges such as technical jargon, implicit assumptions, and regulatory ambiguity.
- **Optimized multi-stage query refinement and enhanced retrieval.** We design a multi-stage refinement pipeline—incorporating term explanation and context injection—that improves query understanding and retrieval precision. This is further integrated with the RAPTOR framework (Sarthi et al. 2024) to support robust, multi-hop retrieval across lengthy regulatory documents.
- **Demonstrated accuracy and real-world applicability.** Extensive evaluation shows substantial improvements over baseline LLM and RAG methods in both retrieval accuracy and compliance reasoning, enabling trustworthy, explainable outputs for real-world deployments.

Related Work

Large Language Models

Large Language Models (LLMs) such as ChatGPT (Int 2024), Gemini (Team et al. 2023), and Qwen (Yang et al. 2025) have demonstrated remarkable success across a wide range of natural language processing (NLP) tasks. Built on the transformer architecture, LLMs are typically pre-trained on web-scale text corpora with billions of parameters, enabling strong language understanding, generation, and transferability to diverse downstream tasks. More recently, "think-first" models with enhanced reasoning capabilities (Guo et al. 2025; Yang et al. 2025) have shown promise in handling complex user requests, decision-making, and external tool invocation, opening new possibilities for fully automated workflows in industrial applications.

Despite these advances, general-purpose LLMs remain vulnerable to hallucinations (Xu, Jain, and Kankanhalli 2025) and often lack the specialized domain knowledge required for enterprise deployment. To address these limitations, various strategies have been explored, including fine-tuning on domain-specific data and retrieval-augmented generation (RAG), which enhances factual reliability and contextual grounding for industry-specific use cases.

Retrieval-Augmented Generation (RAG)

Recently, Retrieval-Augmented Generation (RAG) (Lewis et al. 2021) has emerged as a practical alternative to fine-tuning for adapting LLMs to domain-specific tasks. By retrieving relevant context from external knowledge bases during inference, RAG enhances LLM performance while

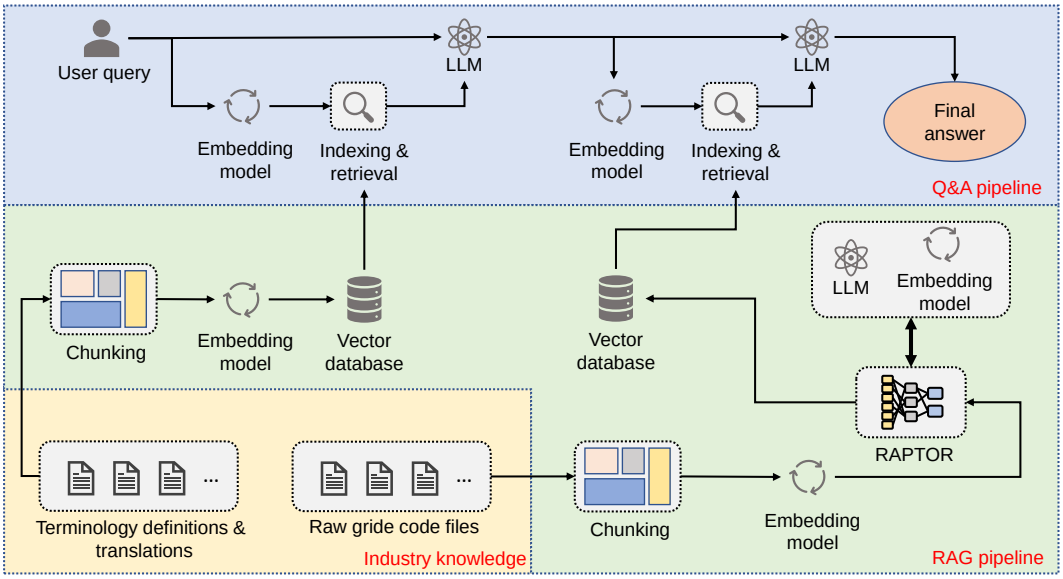

Figure 2: System architecture of GridCodex: The workflow consists of three major components—industry knowledge, a retrieval-augmented generation (RAG) pipeline, and a Q&A pipeline. Domain-specific terminologies and factual knowledge are processed, embedded, and indexed to construct industry knowledge bases. These knowledge bases support multi-stage query refinement and enable robust compliance interpretation in the Q&A pipeline.

avoiding costly and time-intensive model retraining. Advances such as RAPTOR (Sarthi et al. 2024) and HyDE (Gao et al. 2022) have further improved retrieval quality for complex document understanding, achieving notable gains in long-context domains such as legal and scientific texts. With its ability to improve factual consistency, reduce hallucinations, and offer greater controllability, RAG is particularly well-suited for knowledge-intensive applications—including the interpretation and compliance verification of power grid codes.

LLM applications in the Energy Sector

Despite the growing interest in LLMs and RAG, their adoption in the power and energy sector remains at an early stage. Recent studies have explored their use in power system question answering—for instance, in dispatch control (Zhang et al. 2024) and general Q&A tasks (Ni et al. 2024; Lu et al. 2024). However, far less attention has been given to applying LLMs for regulatory interpretation and grid code compliance, a domain where accuracy and reliability are paramount. Existing approaches predominantly rely on supervised fine-tuning to adapt general-purpose LLMs, with limited exploration of model selection and retrieval optimization strategies. To the best of our knowledge, no prior work has systematically employed a RAG-based framework for power grid code reasoning—highlighting a critical gap in both research and practice.

System Architecture

To enable robust grid code reasoning and compliance verification, GridCodex integrates a tailored workflow that leverages RAG to enhance the accuracy of general-purpose LLMs when responding to grid code queries. As illustrated in Figure 2, the system is organized into three core components: industry knowledge, the RAG pipeline, and the Q&A pipeline. The *industry knowledge* module provides domain-specific terminology, definitions, and translations, enabling LLMs to better interpret user queries. The *RAG pipeline* constructs domain knowledge bases and performs retrieval from external regulatory documents. Finally, the *Q&A pipeline* interfaces directly with users, orchestrating retrieved context and LLM reasoning to produce reliable and contextually grounded answers.

Industry Knowledge

Industry knowledge (or domain knowledge) is essential for the effective deployment of LLMs in industrial applications. A key limitation of general-purpose LLMs is their lack of specialized expertise, which prevents them from consistently generating accurate answers to domain-specific queries. In the context of grid code compliance, domain knowledge can be broadly classified into two categories: (1) terminology knowledge, including definitions and translations of technical terms, and (2) factual knowledge, consisting of regulatory clauses and provisions from different countries. In GridCodex, these two knowledge types are maintained as separate knowledge bases and directly supplied to the RAG pipeline, eliminating the need for additional preprocessing while ensuring clarity and modularity.

RAG pipeline

We construct our RAG pipeline using RAGFlow, an open-source framework chosen for its scalability and flexibility.

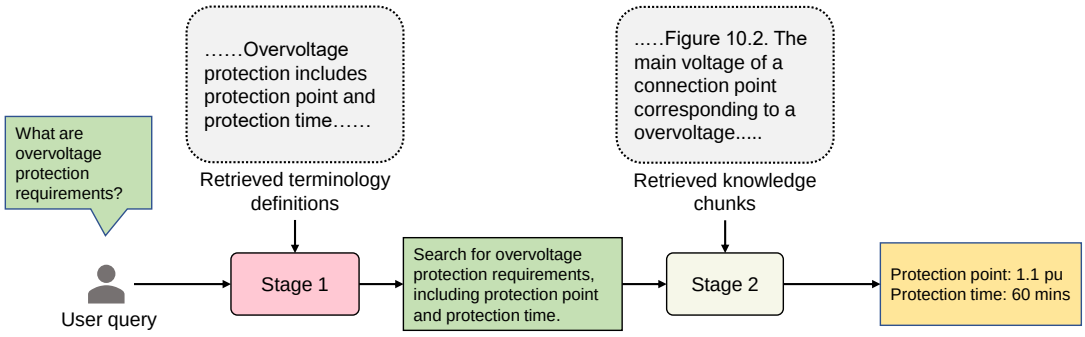

Figure 3: A real use case of GridCodex: In Stage 1, the user query is refined with terminology definitions, then used for raw knowledge retrieval to obtain the final answer.

To maximize retrieval accuracy, we process terminology knowledge and factual knowledge differently.

For terminology definitions and translations, we preserve their hierarchical relationships by storing them in JSON and Markdown formats. These files are chunked to match the embedding model's maximum context window, embedded into dense vectors, and indexed in a vector database.

For factual data—including tables, figures, and raw grid code documents—we employ OCR (Optical Character Recognition), TSR (Table Structure Recognition), and DLR (Document Layout Recognition) as preprocessing steps. Grid codes often contain deeply nested hierarchical structures (e.g., sub-clauses), which pose challenges for retrieval. To preserve semantic integrity, we adopt an adaptive chunking strategy, using the lowest-level section titles as atomic units and preventing clause cutoffs during retrieval. To further capture semantic connections and cross-references, we integrate RAPTOR (Sarthi et al. 2024). RAPTOR leverages Gaussian Mixture Models to cluster semantically related chunks, which are then summarized by general-purpose LLMs. These summaries are re-embedded recursively, producing a tree-structured knowledge base that preserves both local detail and global context. The final knowledge bases, covering both terminology and factual data, are stored in vector databases for efficient retrieval.

Q&A Pipeline

Our Q&A pipeline follows a classic RAG workflow, in which retrieved knowledge is appended to user queries as contextual input. To improve retrieval accuracy, we employ a multi-stage query refinement strategy. First, terminology knowledge is retrieved and used to enrich the original query with detailed explanations and relevant domain-specific keywords. The refined query is then translated into English—aligning with the predominant language of grid code documents—before being applied to factual knowledge retrieval. This staged rewriting process leverages key terminologies, bridges cross-lingual gaps, and ensures more accurate and robust retrieval. The resulting knowledge, together with the refined English query, is then passed to general-purpose LLMs for answer generation. To further enhance reliability, we incorporate rigorous prompt engineering to guide model reasoning toward precise and truthful responses. Importantly, the Q&A pipeline is model-agnostic: it can operate with both API-based LLMs and self-hosted open-source models, offering flexibility for different deployment environments.

Experiments

Experiment Setup

We evaluate *GridCodex* against two baselines:

- **Grid code compliance with general LLMs:** Queries are directly submitted to general-purpose LLMs with minimal prompt engineering and region-specific metadata. Guardrails are included to suppress hallucinations when the model encounters queries outside its knowledge.
- **Grid code compliance with vanilla RAG:** This baseline incorporates RAPTOR-enabled knowledge bases (Sarthi et al. 2024). Questions are used to retrieve relevant chunks, which are appended to the user queries before being processed by a general LLM. Prompt-level guardrails are added to mitigate hallucinations when retrieval fails.

We evaluate system performance on three key metrics:

- **Answer Quality:** It evaluates *accuracy* (correctness), *completeness* (coverage of essential points), and *usefulness* (relevance to the query). Accuracy is prioritized, with completeness and usefulness downweighted when accuracy is low.
- **Faithfulness:** It measures consistency between the generated answers and the retrieved knowledge chunks.
- **Recall@30:** It assesses whether the information necessary to answer a query is contained within the top 30 retrieved chunks.

To ensure fairness and objectivity, we adopt automated scoring using LLM-based evaluators. Carefully designed prompts and expert-provided reference answers from the electricity industry guide the evaluation.

Datasets

We benchmark *GridCodex* using a proprietary HUAWEI grid code dataset comprising question–answer pairs derived from official documents issued by four regulatory agencies:

Table 2: Benchmark results for *GridCodex* and baselines.

Region	Model	Answer Quality	Faithfulness	Recall@30
Hong Kong, China	General LLMs	0.798	–	–
	Vanilla RAG	0.759	0.978	0.182
	Optimized RAG (*GridCodex*)	**0.946**	**0.978**	**1.000**
Netherlands	General LLMs	0.602	–	–
	Vanilla RAG	0.519	**0.978**	0.100
	Optimized RAG (*GridCodex*)	**0.852**	0.968	**0.917**
European Union	General LLMs	0.602	–	–
	Vanilla RAG	0.645	0.941	0.100
	Optimized RAG (*GridCodex*)	**0.843**	**0.968**	**0.913**
Bangladesh	General LLMs	0.784	–	–
	Vanilla RAG	0.805	**0.984**	0.000
	Optimized RAG (*GridCodex*)	**0.877**	0.962	**0.900**

Hong Kong (China), the Netherlands, the European Union, and Bangladesh. The dataset contains 148 QA pairs in total, covering diverse scenarios of grid code interpretation and compliance verification.

Models

The *GridCodex* integrates carefully selected LLMs and embedding models, chosen to balance retrieval accuracy, reasoning ability, answer quality, and computational efficiency:

- **Embedding model:** `Linq-Embed-Mistral` (Choi et al. 2024) is used to construct vector embeddings of knowledge chunks. Trained on high-quality synthetic data, it delivers strong precision and reliability in semantic search.
- **RAPTOR summarization:** `DeepSeek-R1-0528` (Guo et al. 2025) is employed for hierarchical document summarization, leveraging strong reasoning capabilities for semantic clustering.
- **Query refinement and translation:** For terminology expansion and translation, we adopt `Qwen3-32B-AWQ` (Yang et al. 2025), which balances refinement quality and hardware efficiency.
- **Answer synthesis and scoring:** `Qwen3-235B-A22B` (Yang et al. 2025), a mixture-of-experts (MoE) model, is used for final answer generation and automated scoring, providing concise, faithful responses.

Results

Table 2 presents the results across four regulatory regions. Since the plain LLM baseline lacks retrieval, only answer quality is reported. Vanilla RAG provides moderate improvements in certain cases (e.g., the EU and Bangladesh datasets), but its performance is inconsistent, particularly for Hong Kong and the Netherlands. These deficiencies are largely attributable to language mismatches and terminology ambiguities: vanilla RAG frequently retrieves noisy or partially relevant chunks, reflected in its low Recall@30 scores.

In contrast, *GridCodex* consistently outperforms both baselines across all evaluation metrics. Through multi-stage query refinement and RAPTOR-based retrieval, it achieves high and stable document coverage (Recall@30 $>$ 0.90 in all regions) while generating accurate and faithful answers. For example, in the Hong Kong (China) dataset, *GridCodex* achieves perfect retrieval coverage (Recall@30 = 1.0) and a remarkable answer quality score of 0.946, yielding an overall F1 of 0.972.

Similar gains are observed for the Netherlands and EU datasets, which are more challenging due to complex, data-heavy documents. Here, *GridCodex* improves answer quality by more than 30% relative to vanilla RAG, while maintaining strong faithfulness ($\sim$0.968) and high retrieval coverage (Recall@30 = 0.917 and 0.913, respectively). For Bangladesh, where the grid code documents are shorter but contain under-defined clauses, *GridCodex* still yields significant improvements, raising answer quality from 0.805 to 0.877 with Recall@30 maintained at 0.900.

As shown in Figure 4, *GridCodex* delivers an overall 27.5% improvement in answer quality compared to the two baselines. These gains stem from its multi-stage query refinement and RAPTOR-enhanced multi-hop retrieval. The framework also achieves high faithfulness through careful model selection and prompt engineering. Most notably, *GridCodex* increases Recall@30 by nearly 9$\times$ compared with the vanilla RAG system, establishing itself as a robust end-to-end solution for grid code reasoning and compliance.

Ablation Study: Model Size and Reasoning Capability

We conduct an ablation study to assess how model size and reasoning capability influence *GridCodex* performance. All experiments are performed on the Netherlands dataset using the default *GridCodex* configuration. To isolate the effect of the LLMs, we keep query refinement, embedding models, and RAPTOR settings fixed across comparisons.

Model Size: Large vs. Mid-size

To study the role of scale, we compare: (1) `Qwen3-235B-A22B` (Yang et al. 2025), the flagship model with 235 billion total parameters and 22 billion active, optimized for long-context reasoning, and (2) `Qwen3-30B-A3B` (Yang et al. 2025), a 30B-parameter

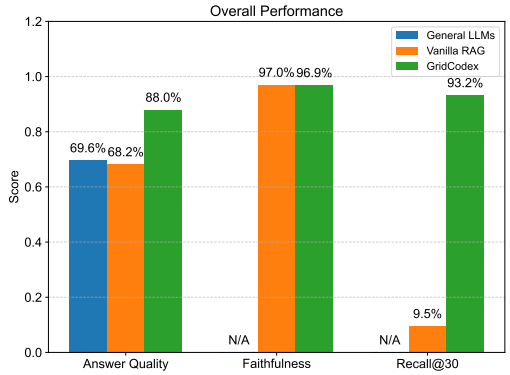

Figure 4: Overall performance of different systems across regions. (Subject to aesthetic changes.)

variant with 3B active, designed for faster inference and reduced hardware costs.

Table 3: Model size impact (Netherlands dataset).

Model	Answer Quality	Faithfulness
235B-A22B	**0.852**	0.968
30B-A3B	0.740	**0.995**

As shown in Table 3, scaling from 30B to 235B parameters improves answer quality by 11.2%. The larger model demonstrates stronger multi-hop reasoning, cross-clause referencing, and ambiguity resolution due to greater representational capacity and expert routing. However, this gain comes at a cost: faithfulness drops slightly (0.968 vs. 0.995), as the 235B model integrates retrieved context more aggressively, occasionally introducing inferred connections not explicitly present in the documents. This shows that larger models enable richer reasoning but may lean toward speculative integration, whereas smaller models remain more conservative and strictly grounded in retrieved evidence.

Reasoning Capability: Thinking vs. Non-Thinking

Next, we examine the effect of reasoning ability by comparing two variants of the same 235B model: (1) `Qwen3-235B-A22B` (Yang et al. 2025), the default reasoning-capable flagship, and (2) `Qwen3-235B-A22B-Instruct-2507` (Yang et al. 2025), an instruction-following variant optimized for direct answer generation with reasoning chains disabled.

Table 4: Impact of reasoning optimization (Netherlands dataset).

Model	Answer Quality	Faithfulness
235B	**0.852**	0.968
235B-Instruct	0.795	**0.989**

Table 4 shows that the reasoning-enabled model delivers a 5.8% improvement in answer quality. Reasoning capability proves essential for deep contextual linking, enabling the model to resolve implicit dependencies and navigate dense regulatory cross-references. Conversely, the Instruct variant, while slightly weaker in reasoning, achieves higher faithfulness by adhering more strictly to retrieved evidence.

Key Observations

The ablation highlights two major findings:

1. **Model scale** boosts reasoning and ambiguity resolution but can introduce speculative interpretations that reduce strict faithfulness.
2. **Reasoning ability** enhances clause linking and implicit dependency resolution, albeit with a tendency toward more interpretive answers.

For real-world deployments, mid-sized reasoning models (e.g., `Qwen3-30B-A3B`) represent a strong balance—retaining much of the reasoning power of very large models while being computationally efficient.

Lessons Learned

From developing and evaluating *GridCodex*, we distill several practical insights:

- **Better queries lead to better answers.** Grid codes contain ambiguous references, nested clauses, and dense jargon, making both retrieval and QA challenging. Direct retrieval on raw user queries often fails to capture the most relevant knowledge chunks, yielding incomplete context. Moreover, without explicit clarification of key terms, LLMs struggle to interpret queries or fully leverage retrieved knowledge. Our multi-stage query refinement—combining terminology explanation, rewriting, and translation—significantly improves retrieval coverage and overall answer quality.
- **Regulatory QA requires reasoning beyond surface text.** Regulatory documents are not simple fact repositories; they demand reasoning across implicit dependencies and extensive cross-references. Our experiments show that reasoning-oriented models consistently outperform purely instruction-following models, underlining the necessity of reasoning ability for compliance interpretation.

These lessons suggest that effective grid code reasoning systems require careful integration of algorithmic design (retrieval and reasoning) with engineering optimizations (query refinement and model selection).

Conclusion

This paper presented *GridCodex*, a retrieval-augmented generation (RAG) framework designed for reasoning and compliance verification in power grid codes. By combining domain-specialized knowledge bases, multi-stage query refinement, and high-performance open-source LLMs, *GridCodex* delivers significant gains in both answer quality and reliability. Extensive experiments demonstrate that *GridCodex* effectively interprets technical terminologies, nested clauses, and complex cross-references, enabling accurate grid code reasoning and compliance assessment. Beyond power grids, the framework shows strong potential for

proactive violation detection, automated infrastructure configuration, and broader compliance workflows in other safety-critical domains.

References

2016. Grid Code Reinforcements for Deeper Renewable Generation in Insular Energy Systems. *Renewable and Sustainable Energy Reviews*, 53: 163–177.

2017. Renewables Integration on Islands. In *Renewable Energy Integration*, 319–329. Academic Press.

2024. Introducing ChatGPT. https://openai.com/index/chatgpt/.

Balaguer, A.; Benara, V.; Cunha, R. L. d. F.; Filho, R. d. M. E.; Hendry, T.; Holstein, D.; Marsman, J.; Mecklenburg, N.; Malvar, S.; Nunes, L. O.; Padilha, R.; Sharp, M.; Silva, B.; Sharma, S.; Aski, V.; and Chandra, R. 2024. RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture. arXiv:2401.08406.

Chen, Y.; Qian, S.; Tang, H.; Lai, X.; Liu, Z.; Han, S.; and Jia, J. 2024. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. arXiv:2309.12307.

Choi, C.; Kim, J.; Lee, S.; Kwon, J.; Gu, S.; Kim, Y.; Cho, M.; and Sohn, J.-y. 2024. Linq-Embed-Mistral Technical Report. arXiv:2412.03223.

Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; and Li, Q. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. arXiv:2405.06211.

Gao, L.; Ma, X.; Lin, J.; and Callan, J. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. arXiv:2212.10496.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; and Liu, T. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2): 1–55.

J, M. R.; VM, K.; Warrier, H.; and Gupta, Y. 2024. Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations. arXiv:2404.10779.

Jeong, C. 2024. Fine-Tuning and Utilization Methods of Domain-specific LLMs. *Journal of Intelligence and Information Systems*, 30(1): 93–120.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report.

Lu, Y.; Peng, J.; Xu, X.; He, Y.; Li, T.; Wei, J.; Jing, H.; Wang, H.; Xu, B.; and Song, H. 2024. 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*, 95–100. Chengdu China: ACM. ISBN 979-8-4007-1019-3.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2025. Large Language Models: A Survey. arXiv:2402.06196.

Ni, M.; Zhang, J.; Fu, C.; Wang, J.; Ning, X.; and Li, S. 2024. ChatGrid: Intelligent Knowledge Q&A for Power Dispatching Control Based on Large Language Models and Retrieval-augmented Generation. In *2024 IEEE 7th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 7, 921–925.

Ntomaris, A. V.; Bakirtzis, E. A.; Chatzigiannis, D. I.; Simoglou, C. K.; Biskas, P. N.; and Bakirtzis, A. G. 2014. Reserve Quantification in Insular Power Systems with High Wind Penetration. In *IEEE PES Innovative Smart Grid Technologies, Europe*, 1–6.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; and Manning, C. D. 2024. RAPTOR: RECURSIVE ABSTRACTIVE PROCESSING FOR TREE-ORGANIZED RETRIEVAL.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ullah, M.; Guan, Y.; Yu, Y.; Chaudhary, S. K.; Vasquez, J. C.; and Guerrero, J. M. 2025. Dynamic Performance and Power Quality of Large-Scale Wind Power Plants: A Review on Challenges, Evolving Grid Code, and Proposed Solutions. *IEEE Open Journal of Power Electronics*, 6: 1148–1173.

Xu, Z.; Jain, S.; and Kankanhalli, M. 2025. Hallucination Is Inevitable: An Innate Limitation of Large Language Models. arXiv:2401.11817.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report.

Zhang, K.; Li, L.; Xu, X.; Xin, R.; Xu, X.; and Zhang, P. 2024. Intelligent Q&A System for Power Dispatching Based on Retrieval Augmented Generation. In *2024 6th International Conference on Energy, Power and Grid (ICEPG)*, 1604–1608.