GCIG: GraphRAG-based Cross-document Instruction Generation for Boosting LLM Reasoning

Xiaoliang Xu, Huang Yuan, Junmei Wang*, Can Xu
School of Computer Science, Hangzhou Dianzi University, Hangzhou, China
xxl@hdu.edu.cn, yuanh@hdu.edu.cn, jmwang@hdu.edu.cn, xucan@hdu.edu.cn

Abstract

Automatic instruction generation offers a low-cost, high-efficiency pathway for fine-tuning large language models (LLMs). However, existing methods struggle in knowledge-intensive domains and complex reasoning tasks due to their dependence on high-quality seed data, limited coverage of single-document knowledge, and repetitive content. To overcome these limitations, this paper presents GCIG, a GraphRAG-based Cross-document Instruction Generation framework. We begin by constructing an enhanced knowledge graph to provide a structural representation of the raw corpus, followed by LLM-driven selection of reliable subgraph-text pairs based on factuality and logical complementarity. Subsequently, we adaptively generate diverse questions through task-aware prompts and context-sensitive retrieval. Finally, we employ Chain-of-Thought reasoning to anchor entity paths and integrate scattered evidence, thereby closing logical gaps and improving answer coherence. Experiments on knowledge-intensive and multi-hop question-answering tasks demonstrate that GCIG outperforms existing methods, producing instruction data with stronger logical consistency and broader knowledge coverage for effective LLM fine-tuning. The code and data are publicly available at https://github.com/WhitEiller/GCIG.

1 Introduction

In recent years, large language models (LLMs) have demonstrated strong capabilities in natural language understanding, generation (Yang et al., 2025a; Jiang et al., 2024), zero-shot learning (Nayak et al., 2024), and complex reasoning (Wei et al., 2022). To better align with human intent, instruction fine-tuning is often essential (Ouyang et al., 2022; Zhang et al., 2023). Its effectiveness, however, fundamentally depends on the quality of the instruction data—especially in high-stakes domains like healthcare, law, and finance, where diverse, domain-consistent, and logically coherent data is critical for ensuring model reliability and specialized performance (Li et al., 2024; Huang et al., 2024).

Due to the high cost of manual annotation, automated instruction generation has become a key approach for constructing instruction data. Representative methods such as Self-Instruct (Wang et al., 2023) and WizardLM (Xu et al., 2024) leverage a general-purpose LLM to generate large-scale instructions from limited seed tasks or unstructured corpora, thereby providing scalable supervision for instruction tuning (Honovich et al., 2023; Ding et al., 2023). However, these methods typically lack structured organization of the source corpus and do not model cross-document relationships. As illustrated in Figure 1(a), they often simply split the corpus into isolated text blocks and generate instructions per block independently. This results in instructions with limited source variety, low diversity, and insufficient knowledge coverage, making them inadequate for complex reasoning tasks such as multi-hop Question and Answer (Q&A). Specifically, three main issues arise:

- **Dependence on high-quality seeds.** Current instruction-generation methods (e.g., Self-Instruct (Wang et al., 2023)) require diverse, high-quality seed instructions to begin. In knowledge-intensive domains, however, building such a broad-coverage seed set is particularly difficult. Even approaches that start from unlabeled text (e.g., Bonito (Nayak et al., 2024)) remain limited by the distribution of their initial data.
- **Limited coverage of single-document knowledge.** Existing automatic instruction generation methods (e.g., Bonito (Nayak et al., 2024)) typically operate on isolated

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 17593–17609

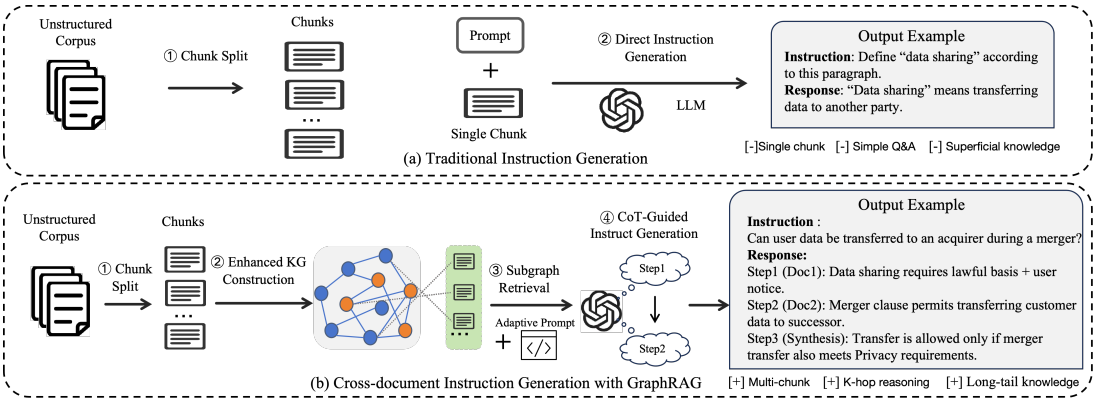


Figure 1: Overview of two paradigms for instruction synthesis. (a) Conventional pipelines generate instructions from isolated text chunks, which limits cross-document evidence integration and multi-hop reasoning coverage. (b) GCIG performs graph-grounded cross-document synthesis with an Enhanced KG, producing instructions with traceable reasoning paths and stronger evidence connectivity.

documents or segments, increasing complexity via expansion or local rewriting. Without cross-document integration, however, the resulting instructions remain simplistic and unsuitable for multi-hop reasoning.

- **Repetitive content and generalization decay.** Static instruction generation templates produce homogeneous content with uniform difficulty levels (Gudibande et al., 2023; Shumailov et al., 2023). Consequently, the resulting low-diversity data can cause LLMs to overfit and impair their generalization in reasoning tasks.

Although recent methods such as RAG-Instruct (Liu et al., 2025) augment generation with retrieved knowledge, they primarily rely on vector-based similarity matching. This retrieval paradigm is effective for identifying locally relevant passages. However, it is less well suited to connecting semantically distant but logically related evidence across documents. Such connections are central to multi-hop reasoning. Other studies, such as EntiGraph (Yang et al., 2025b), introduce graph-structured representations to better capture cross-document knowledge topology through entity linking. However, existing approaches tend to emphasize either retrieval quality or graph organization. They do not explicitly preserve and propagate fine-grained evidence links throughout downstream retrieval and generation. As a result, it remains difficult to ensure that multi-hop instructions are both logically connected and grounded in traceable supporting evidence.

To tackle this challenge, we propose a GraphRAG-based method for automatically generating cross-document instruction data to enhance LLMs' complex reasoning capabilities. As illustrated in Figure 1(b), we first build an enhanced Knowledge Graph (KG) linking document evidence with entities. The **LLM-driven Chunk Selection (LCS)** module then forms multi-hop evidence chains. Using these chains, the **Adaptive Prompt-based Question Generation (APQG)** module produces multi-hop questions, and the **Chain-of-Thought-guided Answer Generation (CoTAG)** module generates interpretable answers, yielding high-quality instruction data automatically.

Specifically, our major contributions are:

- We propose a GraphRAG-based method for automatically synthesizing high-quality instructions. Evaluation shows its efficacy for domain reasoning and multi-hop reasoning.
- We construct an Enhanced KG to structure the raw corpus, and employ LCS to quantitatively evaluate the retrieved subgraph-text pairs for factuality and logical complementarity, ensuring a reliable knowledge source for high-quality instruction generation.
- APQG ensures broad instruction distribution. By anchoring entity paths and integrating unstructured evidence via CoTAG, it addresses logical gaps in answers and improves depth and reliability.

2 Related Work

Instruction Tuning Instruction tuning has become a central paradigm for eliciting the general-

ization capabilities of LLMs (Yang et al., 2025b). Early studies, including FLAN (Chung et al., 2024; Wei et al., 2021) and T0 (Sanh et al., 2022), showed that increasing task diversity during fine-tuning substantially improves zero-shot performance. Later, Self-Instruct (Wang et al., 2023) and Alpaca (Taori et al., 2023) reduced the cost of data construction by using capable models to bootstrap instruction data. Bonito (Nayak et al., 2024) further formulates instruction synthesis as "conditional task generation." This formulation enables the generation of instructions with controllable task types directly from unlabeled domain text and supports zero-shot task adaptation. To increase instruction complexity, WizardLM (Xu et al., 2024) and the Orca (Mukherjee et al., 2023) series introduced instruction evolution and the imitation of detailed reasoning traces, respectively. These methods improve the cognitive density of the resulting training data. However, most existing approaches rely primarily on single documents or general-domain knowledge during generation. As a result, they lack systematic mechanisms for organizing evidence in scenarios that require cross-document evidence integration. This limitation often leads to logically flattened instructions or factual hallucinations.

Retrieval-augmented generation Retrieval-augmented generation (RAG) enhances LLMs with external knowledge and helps reduce hallucinations (Gao et al., 2023; Asai et al., 2024). However, traditional vector-based retrieval often lacks global context. This weakness can produce results that are locally relevant but globally inconsistent, especially for long-tail queries and multi-step reasoning (Sciavolino et al., 2021). GraphRAG addresses this limitation by leveraging knowledge graphs and community detection to integrate information in a hierarchical manner (Edge et al., 2024; Traag et al., 2019). Despite these advances, achieving high accuracy in domain-specific settings with LLMs remains a persistent challenge.

Domain-Specific Challenges In knowledge-intensive domains such as medicine (Savage et al., 2025), biology (Chen et al., 2023), and law (Guha et al., 2023), general-purpose LLMs often fail to meet stringent accuracy requirements due to a deficiency in specialized knowledge. Current research focuses on injecting domain knowledge into models through tailored strategies (Song et al., 2025; Gururangan et al., 2020); for instance, InstructProtein (Wang et al., 2024) achieves alignment between human and protein languages by constructing knowledge-centric instructions. Although incorporating knowledge graphs to aid reasoning has improved model interpretability, existing methods are often limited to the simple retrieval of isolated knowledge points. They struggle to handle complex logic that necessitates cross-document integration, multi-hop reasoning, and the accommodation of exceptional cases.

3 Method

This section introduces the GraphRAG-based Cross-document Instruction Generation (GCIG) framework, which aims to automate the generation of high-quality instructional data for domain-specific knowledge reasoning and multi-hop question answering. As depicted in Figure 2, the framework follows a structured four-stage workflow. **Step 1**. Enhanced KG Construction (§3.1); **Step 2**. LLM-driven Chunks Selection (§3.2); **Step 3**. Adaptive Prompt-based Question Generation (§3.3); **Step 4**. Chain-of-Thought (CoT)-Guided Answer Generation (§3.4).

Given a domain-specific raw text corpus $D = \{d_i\}$, the goal of the framework is to automatically generate high-quality instruction-tuning samples $I = \{C_i, Q_j, A_j\}$, where C_i represents the context retrieved and filtered from D, Q_j denotes a question generated conditioned on C_i, and A_j is the corresponding answer derived from the context.

3.1 Enhanced KG Construction

To fit within LLM input length limits, we first chunk the documents into coherent segments ($\leq$ 2048 tokens). These segments form a domain-specific knowledge graph, from which entities and relations are extracted using predefined types—covering general categories (e.g., dates, locations) and domain-specific terms. Mentions of the same entity or relation across segments are merged into a unified graph $G = \{E, R\}$, which is then semantically enriched to support cross-document reasoning.

Furthermore, semantic enhancement is performed on the knowledge graph G to obtain an enriched structure tailored for cross-document reasoning. Specifically, each entity and relation is explicitly linked to its originating sentence-level contexts, resulting in triples of the form $\langle$entity, relation, context$\rangle$. After semantic consolidation and context binding, we obtain the

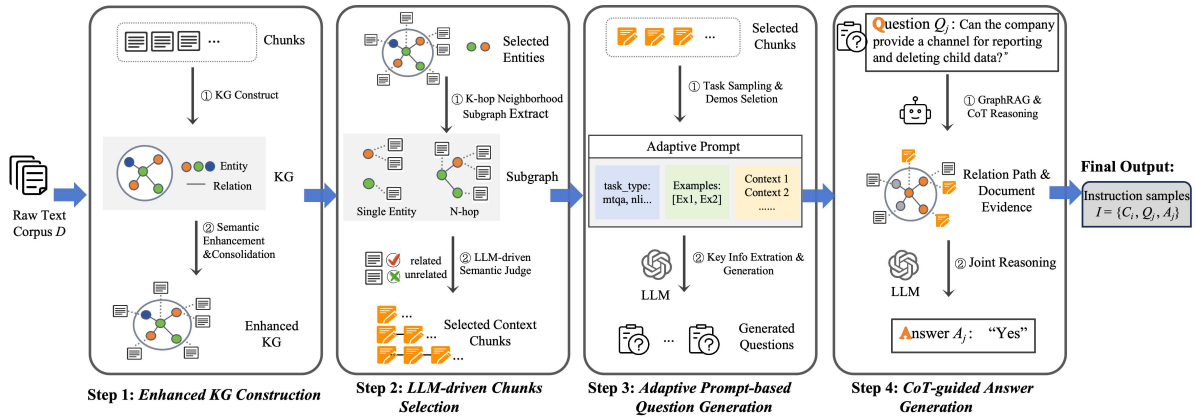


Figure 2: Detailed overview of GCIG. The framework comprises four stages: (a) Enhanced KG construction, which builds a semantically enriched graph with entity–relation–context bindings from raw corpora; (b) LLM-driven chunk selection, which extracts K-hop subgraphs and filters evidence by factual consistency and logical complementarity; (c) adaptive prompt-based question generation, which instantiates task-aware prompts from selected graph-grounded contexts; and (d) CoT-guided answer generation, which enforces graph-constrained, evidence-traceable reasoning to produce final instruction tuples.

semantically enhanced knowledge graph $G = \{E, R, C\}$. Unlike previous approaches, our implementation retains structured domain knowledge while enabling fine-grained evidence tracing and interpretable reasoning over context fragments. This method establishes a solid knowledge foundation for downstream GraphRAG-based retrieval, multi-hop cross-document QA, and instruction generation.

3.2 LLM-driven Chunks Selection

To capture the multi-granular semantics of entities, from local attributes to global associations, we first extract a K-hop neighborhood subgraph centered on the core entities. This subgraph covers the 0-hop, 1-hop, and n-hop scopes; see Algorithm 1. During extraction, we use beam search for layer-wise expansion under two constraints. One constraint limits the topological depth to at most K hops. The other constrains the total text length after linearization, for example, with a predefined token budget. These constraints allow the retrieved subgraphs to achieve broad structural coverage while remaining within the context window of the LLM.

Building on this structured representation, we further introduce an LLM-driven semantic consistency discrimination mechanism. This mechanism identifies high-quality subgraph–text pairs. It uses the reasoning capabilities of LLMs to evaluate the alignment between candidate text chunks and subgraph structures in a quantitative manner. The evaluation considers two aspects: factual consistency and logical complementarity. Detailed scoring criteria, aggregation methods, and filtering rules are provided in Appendix C. This process reduces semantic noise effectively. The resulting subgraph–text pairs combine the explicit structural constraints of the knowledge graph with the implicit reasoning capabilities of LLMs. They thus provide reliable contextual grounding for high-quality instruction generation.

Algorithm 1 K-hop Subgraph Extraction

Input: Graph G, Anchor entity e_{core}, hop limit k, beam width b, token budget L
Output: Subgraph G'

$G' \leftarrow \emptyset$; $E \leftarrow \{e_{core}\}$; $h \leftarrow 0$
while $h < k$ **and** $E \neq \emptyset$ **and** $\text{TokenLen}(\text{Linearize}(G')) < L$ **do**
  $C \leftarrow \text{AdjEdges}(G, E)$
  $\mathcal{R} \leftarrow \text{BeamSelect}(C, b, LLM_{score})$
  for each $r \in \mathcal{R}$ **do**
    $G' \leftarrow G' \cup \{r\}$
    $E \leftarrow E \cup \text{Endpoints}(r)$
  end for
  $h \leftarrow h + 1$
end while
return G'

3.3 Adaptive Prompt-based Question Generation

Based on the semantically relevant context set C^*, this stage guides the LLM to generate a structured, knowledge-rich instruction set Q. To cover diverse

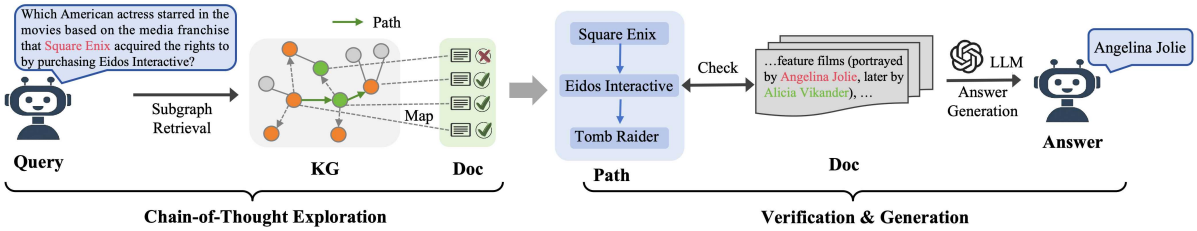


Figure 3: Two-stage workflow of CoT-guided answer generation over the Enhanced KG. In Chain-of-Thought Exploration, the model retrieves a subgraph, explores relation paths, and maps path nodes to supporting documents. In Verification & Generation, candidate path-document evidence is checked, and the LLM produces the final answer from the verified reasoning chain.

task types without supervision, we adopt an adaptive prompting strategy. Unlike static prompts, our approach dynamically binds the task type and the prompt instantiation to C^*, enabling the prompt to adapt to its semantic and structural features. The detailed design of the adaptive prompt template is provided in Appendix D.

Specifically, we first sample a task type τ from the task space $\mathcal{T}$ and instantiate its corresponding template skeleton. Key information relevant to τ—such as entities, relation chains, or constrained context segments—is then extracted from C^* to populate the template slots. Finally, few-shot demonstrations aligned with the task are selected to complete the prompt. The resulting prompt is formalized as follows:

$$\mathcal{P}(C^*, \tau) = \text{Template}(\tau; \mathcal{I}_{\text{fill}}(C^*), \mathcal{E}_\tau(C^*)) \quad (1)$$

where $\text{Template}(\cdot)$ denotes the skeleton of the unified instruction template, $\mathcal{I}_{\text{fill}}(C^*)$ refers to the information units extracted from C^* (e.g., entities or relation chains), and $\mathcal{E}_\tau(C^*)$ represents the few-shot demonstrations selected for task type τ given C^*. This formulation ensures template-level consistency while allowing dynamic adaptation to varying contexts and task demands, thereby improving both the structural stability and the domain-specific validity of the generated instructions.

3.4 CoT-guided Answer Generation

Although the Enhanced KG augments context by linking documents, direct traversal over the expanded search space can lead to semantic drift and fragmented reasoning. To mitigate this, we introduce a CoT-guided answer generation module that explicitly constructs reasoning paths. These paths weave scattered document evidence into a coherent global evidence chain, imposing cross-document constraints to reduce factual inconsistencies while maintaining logical coherence throughout the reasoning process.

As illustrated in Figure 3, the pipeline operates in three stages. First, given a question Q, key entities are identified and anchored to the Enhanced KG to initialize the reasoning process. Next, GraphRAG retrieves relevant subgraphs, during which CoT prompting guides the LLM to construct an explicit relational reasoning path over the graph. This path includes both entity relations and their associated evidentiary documents. Because the Enhanced KG links unstructured documents to graph nodes, the resulting reasoning path not only provides a logical backbone across documents but also serves as a direct index to the supporting texts attached along the path. Finally, the LLM integrates this structured relational path with fine-grained semantic information from the retrieved documents to perform joint reasoning and produce the answer A. The detailed CoT prompt used in the experiments is provided in Appendix E.

The generated question–answer pair, together with the supporting context, constitutes the final instruction sample $I = \{C_i, Q_j, A_j\}$.

4 Experimental Setup and Baselines

Datasets. We utilized Contract-NLI (Koreeda and Manning, 2021) and Privacy-QA (Ravichander et al., 2019) to verify the model's ability to capture fine-grained semantic constraints within specialized domains. We introduced BioASQ (Tsatsaronis et al., 2015) to assess the model's transferability and knowledge integration boundaries in the non-target biomedical domain. Finally, we employed HotpotQA (Yang et al., 2018) to test the model's robustness in multi-document evidence integration and complex multi-hopreasoning.

Base Models. We employ Mistral-7B (Jiang et al.,

Model	Supervision Source	Privacy_QA		Contract_NLI		BioASQ		Avg	
		F1	ACC	F1	ACC	F1	ACC	F1	ACC
Mistral	None	44.1	47.6	31.8	41.3	24.7	27.1	33.5	38.7
	TAPT	46.3	47.7	34.2	42.0	26.2	29.1	35.6	39.6
	Bonito	52.5	55.1	71.9	74.2	40.7	43.4	55.0	57.6
	Self-QA	48.2	49.7	53.2	58.1	29.7	32.5	43.7	46.8
	EntiGraph	49.2	51.8	67.3	68.2	41.1	45.4	52.5	55.1
	GCIG (Ours)	**55.9**	**58.3**	**73.9**	**74.9**	**42.3**	**45.1**	**57.4**	**59.4**
Qwen 2.5	None	40.7	44.5	35.7	42.1	28.9	34.5	35.1	40.4
	TAPT	42.1	44.3	36.5	42.9	29.7	36.5	36.1	41.2
	Bonito	49.3	51.2	66.2	71.3	37.9	42.4	51.1	55.0
	Self-QA	43.2	45.3	56.5	60.0	31.8	34.1	43.8	46.5
	EntiGraph	47.2	51.7	62.3	69.6	40.1	47.4	49.9	56.2
	GCIG (Ours)	**55.1**	**57.6**	**69.7**	**75.7**	**45.2**	**49.2**	**56.7**	**60.8**
LLaMA 2	None	38.7	39.7	36.2	38.1	27.4	29.6	34.1	35.8
	TAPT	38.9	39.7	38.4	39.7	29.2	31.9	35.5	37.1
	Bonito	48.3	**51.3**	66.9	**70.2**	46.1	51.2	53.8	57.2
	Self-QA	42.4	46.3	57.2	60.1	44.2	46.3	47.9	50.9
	EntiGraph	41.3	44.2	61.3	64.2	47.1	49.9	49.9	52.8
	GCIG (Ours)	**48.7**	50.2	**68.7**	70.1	**49.2**	**53.1**	**55.5**	**57.8**

Table 1: Performance comparison of supervision strategies on Privacy_QA, Contract_NLI, and BioASQ using three 7B backbones (Mistral, Qwen2.5, and LLaMA 2).

2024), LLaMA-7B (Touvron et al., 2023), and Qwen2.5-7B (Yang et al., 2025a) as backbones to verify the model-agnostic nature of our approach.

Implementation. Models are fine-tuned for 3 epochs using AdamW with a learning rate of 2×10^{-5}, a batch size of 32, and a maximum sequence length of 2048. Instruction generation strictly follows Section 3.

Baseline. Beyond the standard Zero-shot baseline, which relies solely on prompting without auxiliary data, we compare our method against several adaptation and generation approaches: TAPT (Gururangan et al., 2020), which continues pretraining on domain text; Self-QA (Zhang and Yang, 2023), generating weakly supervised question-answer pairs; Bonito (Nayak et al., 2024), sampling free-form QA data broadly; EntiGraph (Yang et al., 2025b), synthesizing corpora via entity-relation graphs.

5 Experimental Results and Analysis

5.1 Impact of Domain Knowledge

This section empirically evaluates the GCIG framework for domain-specific instruction generation and model fine-tuning. The primary goal is to examine whether the proposed approach significantly enhances the performance of pre-trained models, particularly in domain-specialized tasks. To ensure a fair comparison, all fine-tuning experiments were conducted under identical conditions, with strictly controlled data scales and training budgets. To verify the model-agnostic nature of the method, we selected three base models with distinct architectures for evaluation: Mistral-7B, LLaMA-7B, and Qwen2.5-7B. All models were evaluated on the official test sets corresponding to each task. Experimental results are reported in Table 1. For knowledge reasoning in professional fields (Privacy_QA, Contract_NLI, BioASQ), we adopt the official recommended mainstream evaluation metrics Accuracy and F1.

Results Analysis. As shown in Table 1, GCIG achieved consistent performance gains across all base models and downstream tasks, fully confirming the model-agnostic nature of the method. Specifically:

First, our method outperforms the strong baseline Bonito (e.g., +3.4 F1 on Privacy with Mistral), demonstrating the superiority of structured knowledge over generic synthetic data. Second, GCIG significantly surpasses Self-QA methods (e.g., +13.7 F1 on Mistral), as our structured constraints effectively filter noise and hallucinations. Finally, compared to EntiGraph, GCIG achieves better results (e.g., +7.4 F1 on Contract_NLI), proving its ability to capture complex reasoning logic beyond simple entity-relation traversal.

Model	Method	HotpotQA		
		ROUGE-F	ACC	F1
Mistral	None	8.4	9.7	8.1
	TAPT	9.1	10.6	9.2
	Self-QA	13.1	12.2	10.9
	Bonito	11.2	9.8	8.9
	EntiGraph	21.0	16.9	15.1
	GCIG	**29.9**	**22.7**	**19.3**
Qwen2.5	None	9.1	10.4	9.2
	TAPT	11.2	11.9	10.0
	Self-QA	15.1	13.6	12.0
	Bonito	12.2	9.6	8.6
	EntiGraph	19.4	16.1	14.3
	GCIG	**27.8**	**24.2**	**21.9**
LLaMA 2	None	8.3	8.0	7.1
	TAPT	8.4	8.7	8.0
	Self-QA	12.4	11.2	9.7
	Bonito	11.1	10.2	9.6
	EntiGraph	18.7	15.2	13.2
	GCIG	**27.1**	**21.9**	**17.8**

Table 2: HotpotQA results for different fine-tuning strategies across three 7B backbones (Mistral, Qwen2.5, and LLaMA 2).

5.2 Complex Multi-hop Reasoning

To evaluate the model's ability to perform complex multi-hop reasoning, we conduct the experiments on HotpotQA. It is a popular question-answering dataset specifically designed for tasks that require multi-hop reasoning and multi-document support. An additional metric, ROUGE-F, is added to evaluate the semantic coherence and key information coverage of the answers. Other settings are the same as those in Section 5.1.

Results Analysis. Table 2 presents the results on HotpotQA. Using the Mistral model, our GCIG method achieved a ROUGE-F score of 29.9, which is 8.9 points higher than EntiGraph. This improvement demonstrates the value of explicitly constructing connections between text chunks and entities. By enriching the corpus with knowledge semantics and evidentiary constraints, our approach enables LLMs to achieve superior fine-tuning results.

5.3 Cross-document Knowledge Integration

To verify the effectiveness of GCIG in multi-hop reasoning scenarios, we designed a cross-document information integration experiment. This allows us to assess the reasoning gains achieved when the model combines multiple evidence sources. We conducted experiments using the Hotpot dataset. The key innovation of this experiment lies in partitioning the dataset into four fine-grained subsets based on the number of supporting documents required to answer the question: single-document, two-document ($k = 2$), three-document ($k = 3$), and four-document ($k = 4$). This stratified design allows us to precisely quantify the marginal performance of different methods under varying context complexities.

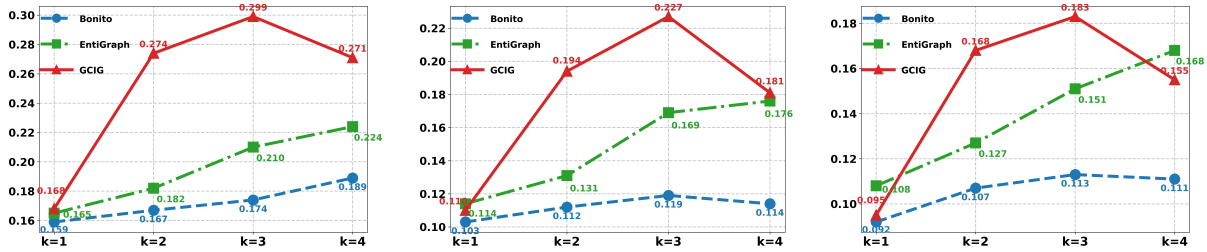
Results Analysis. As shown in Figure 4, Bonito shows clear performance stagnation as the number of documents increases. The model lacks structural guidance. As a result, it struggles to cross document boundaries and connect fragmented clues. This weakness limits effective evidence aggregation during multi-hop reasoning.

Graph-based methods generally outperform Bonito. Among them, GCIG substantially surpasses EntiGraph in the $k = 2$ and $k = 3$ settings. At $k = 3$, the margin reaches approximately 9% in ROUGE-F. However, performance declines in the $k = 4$ setting. This result suggests that a larger document set introduces many additional text chunks. Although these chunks enrich the context, they also bring more distracting information. This makes it harder for the model to focus on the most relevant evidence. The result highlights a key challenge in multi-document settings. More aggressive pruning is needed to preserve core evidence. It is also important to reduce hallucinations and misleading signals from irrelevant context.

5.4 Automatic Evaluation

To compare the quality of instructions synthesized by different methods, we designed a set of task-specific evaluation prompts. We then used DeepSeek-v3 (Liu et al., 2024) as the judge model to perform strict pairwise comparisons between instructions generated by GCIG and those produced by each baseline method.

For each unsupervised source text in every data subset, we generated the same number of instruction sets with GCIG and with the corresponding baseline method. To reduce potential evaluation bias, we anonymized the source of every instruction set before evaluation. We also randomized the presentation order of the compared instruction sets in the judge model input. Based on the judge model's pairwise decisions, we computed the win



(a) ROUGE-F Comparison (b) Accuracy Comparison (c) F1 Score Comparison

Figure 4: Cross-document integration performance on HotpotQA as the number of supporting documents increases ($k = 1, 2, 3, 4$). GCIG outperforms Bonito and EntiGraph at $k = 2$ and $k = 3$, and achieves the best results at $k = 3$ across all three metrics. Performance declines at $k = 4$, suggesting that denser context makes reasoning more difficult.

Dataset	Baseline	GCIG Win Rate (%)	
		Quality	Coverage
Contract_NLI	Bonito	69.4	82.3
	EntiGraph	78.1	74.6
HotpotQA	Bonito	91.7	96.2
	EntiGraph	83.2	79.1

Table 3: Pairwise automatic evaluation with DeepSeek-v3 as the judge model. Values report GCIG win rates (%) against Bonito and EntiGraph on instruction quality and information coverage.

Model	Method	Accuracy	F1	ROUGE-F
Mistral	**Full**	**22.7**	**18.3**	**29.9**
	w/o CoT	21.6 (↓1.1)	17.6 (↓0.7)	28.5 (↓1.4)
	w/o Filter	20.3 (↓2.4)	16.4 (↓1.9)	25.7 (↓4.2)
Qwen2.5	**Full**	**24.2**	**21.9**	**27.8**
	w/o CoT	22.7 (↓1.5)	21.0 (↓0.9)	25.9 (↓1.9)
	w/o Filter	21.4 (↓2.8)	19.7 (↓2.2)	24.3 (↓3.5)

Table 4: Ablation study of GCIG on HotpotQA. **Full** denotes the complete pipeline; *w/o CoT* removes explicit CoT supervision, and *w/o Filter* removes semantic evidence filtering.

rate of GCIG over each baseline. Detailed evaluation prompts and experimental settings are provided in Appendix F.

Results Analysis. As shown in Table 3, GCIG achieves clear advantages on both evaluation metrics. Against Bonito, the win rates on Contract reach 69.4% and 82.3%, respectively. On HotpotQA, the corresponding win rates increase to 91.7% and 96.2%. These results indicate that our method generates information-rich and high-quality instructions more consistently, especially on domain-specific corpora.GCIG also outperforms EntiGraph by a clear margin. In terms of quality, the win rates reach 78.1% on Contract and 83.2% on HotpotQA. For overall information coverage, the corresponding results are 74.6% and 79.1%. These findings suggest that, although EntiGraph expands content through entity relations, it often lacks sufficient contextual support. As a result, the generated instructions can become less precise and less coherent. In contrast, our method explicitly incorporates evidence text chunks associated with relevant entities during generation. This design provides stronger contextual grounding and improves both the logical coherence and the information density of the generated instructions.

5.5 Ablation Studies

To examine the source of GCIG's performance gains and quantify the contribution of its core modules, we conduct ablation studies on the HotpotQA validation set. The experiments address two key questions: (1) whether the multi-document Filter improves evidence quality, and (2) whether CoT elicits deeper reasoning than direct QA mapping.

To ensure fairness, all models were fine-tuned based on Mistral-7B. We compared the full model with the following two variants:

- **w/o Filter:** This variant removes the semantic filter, feeding GraphRAG's raw Top-k retrieval results directly into the model to evaluate its role in suppressing noise and hallucinations.
- **w/o CoT:** Removing only CoT reasoning steps while keeping the filter simplifies the training objective from "Context → Reasoning → Answer" to "Context → Answer," isolating the contribution of explicit reasoning to cross-document logic.

Results Analysis. As shown in Table 4, removing the Filter component leads to the largest accuracy drop, at 2.4%. This result suggests that, although Enhanced GraphRAG provides diverse retrieval paths, noise in the raw retrieval results can still distract the model. The Filter component improves evidence quality through semantic verification. It therefore plays a critical role in reducing hallucinations.

Removing CoT results in a further accuracy drop of 1.1%. This finding confirms its importance. For complex domain-specific problems, simple information mapping is insufficient. CoT encourages the model to construct an explicit reasoning path. As a result, it improves answer consistency.

Overall, these results indicate that the Filter component ensures the quality of the raw corpus, whereas CoT enhances the model's reasoning capacity. Both components are therefore indispensable.

6 Conclusion

We propose GCIG, a GraphRAG-based cross-document instruction generation method for fine-tuning LLMs on knowledge-intensive and multi-hop reasoning tasks. Experiments on three domain-specific datasets show that GCIG can automatically generate high-quality instruction data. The method also alleviates several common issues, including over-reliance on seed instructions, limited knowledge coverage, and content duplication. In addition, results on complex multi-hop reasoning tasks show that GCIG improves the coherence of evidence chains in the generated instructions. This helps reduce factual conflicts while preserving logical completeness. As a result, the generated data better supports the development of LLMs with stronger complex reasoning capabilities. We also observe that incorporating more documents introduces additional distracting text chunks. Although these chunks enrich the context, they can reduce the model's focus on key evidence. This finding highlights an important challenge in multi-document settings.

Future work will focus on two directions. The first is to extend GCIG in a more lightweight manner to reduce computational overhead. The second is to introduce a real-time synergy mechanism and a dynamic feedback loop between graph updating and instruction generation. Such a design may improve the timeliness of evidence chains and support continuous data refinement.

Limitations

Despite its strong performance in GraphRAG-based cross-document instruction generation, GCIG still has several limitations. The framework is effective for knowledge-intensive tasks and complex reasoning. It also reduces over-reliance on seed instructions and preserves coherent evidence chains. However, these benefits involve important trade-offs.

A key limitation is the substantial computational overhead introduced by the framework. Constructing the augmented knowledge graph requires cross-document entity extraction and relation alignment. Both steps are computationally expensive. The challenge becomes more pronounced for large-scale corpora. Consequently, the preprocessing cost may constrain the broader applicability of GCIG. Improving computational efficiency is therefore an important direction for future research.

Another limitation is the absence of a dynamic feedback mechanism. The current framework operates in a one-way manner, from graph construction to instruction generation. It cannot use logical gaps identified during generation to revise the graph afterward. As a result, the knowledge base and the generation model cannot be improved jointly. This limitation reduces the potential for self-refinement and makes iterative optimization of data quality more difficult. Incorporating a feedback-driven refinement process is therefore another important direction for future work.

Acknowledgments

This research is supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant No. 2024C01020 and the Youth Fund of the National Natural Science Foundation of China under Grant No. 62401189.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Qijie Chen, Haotong Sun, Haoyang Liu, Yinghui Jiang, Ting Ran, Xurui Jin, Xianglu Xiao, Zhimin Lin, Hongming Chen, and Zhangmin Niu. 2023. An extensive benchmark study on biomedical text generation and mining with chatgpt. *Bioinformatics*, 39(9):btad557.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428.

Hui Huang, Bing Xu, Xinnian Liang, Kehai Chen, Muyun Yang, Tiejun Zhao, and Conghui Zhu. 2024. Multi-view fusion for instruction mining of large language model. *Information Fusion*, 110:102480.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Wanlong Liu, Junying Chen, Ke Ji, Li Zhou, Wenyu Chen, and Benyou Wang. 2025. Rag-instruct: Boosting llms with diverse retrieval-augmented instructions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3888.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12585–12611.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. 2019. Question

answering for privacy policies: Combining computational and legal perspectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4947–4958, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, and 21 others. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Thomas Savage, Stephen P Ma, Abdessalem Boukil, Ekanath Rangan, Vishwesh Patel, Ivan Lopez, and Jonathan Chen. 2025. Fine-tuning methods for large language models in clinical medicine by supervised fine-tuning and direct preference optimization: Comparative evaluation. *Journal of Medical Internet Research*, 27:e76048.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Zirui Song, Bin Yan, Yuhan Liu, Miao Fang, Mingzhe Li, Rui Yan, and Xiuying Chen. 2025. Injecting domain-specific knowledge into large language models: a comprehensive survey. *arXiv preprint arXiv:2502.10708*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, and 1 others. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Zeyuan Wang, Qiang Zhang, Keyan Ding, Ming Qin, Xiang Zhuang, Xiaotong Li, and Huajun Chen. 2024. Instructprotein: Aligning human and protein language via knowledge instruction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1114–1136.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2025a. Qwen2.5 technical report. *arXiv preprint*, arXiv:2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. 2025b. Synthetic continued pretraining. In *International Conference on Representation Learning*, volume 2025, pages 44379–44421.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, and 1 others. 2023. Instruction tuning for large language models: A survey. *ACM Computing Surveys*.

Xuanyu Zhang and Qing Yang. 2023. Self-qa: Unsupervised knowledge guided language model alignment. *arXiv preprint arXiv:2305.11952*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Dataset Statistics

We evaluate our model on four datasets covering diverse domains: PrivacyQA (Ravichander et al., 2019), ContractNLI (Koreeda and Manning, 2021), BioASQ Task B (Tsatsaronis et al., 2015), and HotpotQA (Yang et al., 2018). Table 5 summarizes the statistics for the training and test splits used in our experiments.

Dataset	Domain	Corpus	Test
PrivacyQA	Privacy Policy	10,923	2,045
Contract_NLI	Legal Contracts	6,819	1,991
BioASQ(B)	Biomedical	8,572	2,000
HotpotQA	Open Domain	12,471	2,000

Table 5: Summary of dataset statistics.

B Instruction Synthesis

Taking into account both performance and computational efficiency, and to mitigate potential biases stemming from specific teacher models, we only use `Qwen3-32B` in our synthesis pipeline.

The pipeline employs a unified prompting procedure across two stages with distinct decoding strategies (Section 3). The first stage, *Instruction Generation*, uses high-entropy sampling to maximize diversity. Conversely, the second stage, *Response Generation*, adopts conservative sampling to ensure faithfulness. Table 6 lists the decoding hyperparameters. We synthesized 20,000 fine-tuning samples per setting.

Hyperparameters	Values
Instruction Generation	
temperature	1.5
top_p	1.0
presence_penalty	0.3
frequency_penalty	0.3
max_tokens	16384
Response Generation	
temperature	0.3
top_p	1.0
presence_penalty	0.5
frequency_penalty	0.5
max_tokens	16384

Table 6: Hyperparameters of Synthetic Task Generation.

C Chunks Selection Scoring Criteria

In this section, we describe the two prompt templates used in the chunk selection stage. Table 9 presents the filtering template for candidate evidence combinations. This template evaluates each candidate text pair using two complementary dimensions: factual consistency, which assesses whether the candidate is grounded in the provided entity–relation evidence, and logical complementarity, which assesses whether the paired texts jointly support a meaningful multi-hop reasoning link. Candidates are retained only when they satisfy the threshold on both dimensions, and are then ranked by their average score.

Table 10 presents the scoring template used for subgraph expansion. This template evaluates candidate edges connected to the starting entity and selects those most beneficial for extending the current subgraph. The evaluation prioritizes four aspects: information gain, to favor novel and non-redundant content; reasoning value, to encourage edges that support multi-hop inference; core connectivity, to preserve semantically central links; and diversity, to avoid repeatedly selecting edges with similar relation or entity patterns.

D Adaptive Prompt Template

In this section, we provide the detailed structure of the adaptive prompting strategy described in the main text. We first illustrate the unified template skeleton used to construct $\mathcal{P}(C^*, \tau)$ in Table 7. Subsequently, we demonstrate how this template is instantiated for two distinct tasks in Table 11 and Table 12, respectively.

The template is designed with modularity in mind. The Instruction slot defines the task boundary, the Few-shot Exemplars ($\mathcal{E}_\tau$) provide in-context guidance specific to the sampled task τ, and the Selected Context ($\mathcal{I}_{\text{fill}}(C^*)$) injects the

grounded information retrieved from the previous stage.

Prompt Template $\mathcal{P}(C^*, \tau)$

Instruction
Generate a {**Task_Type** : τ} based on the text below. The output must strictly follow the format of the examples provided.

Few-shot Exemplars $(\mathcal{E}_\tau)$
Input: $<$ Demo_Context$_1$ $>$
Output: (q_1, a_1)
...
Input: $<$ Demo_Context$_k$ $>$
Output: (q_k, a_k)

Selected Context $(\mathcal{I}_{\text{fill}}(C^*))$
Text: $[[C]; [C^+]]$

Response
(Model generates $q_{\text{new}}, a_{\text{new}}$)

Table 7: The unified adaptive prompt template skeleton.

E CoT Answer Template

In this section, we provide the detailed structure of the CoT answer template described in the main text. We illustrate the full prompt used for CoT-guided answer generation over the Enhanced Knowledge Graph (Enhanced KG) in Table 13.

The template is designed with structured reasoning in mind. The Entity Anchoring stage identifies the relevant entities in the question, the Reasoning Path stage performs step-by-step reasoning over the graph by jointly using entities, relations, and contexts, and the Final Answer stage synthesizes the resulting reasoning chain into a grounded response.

F Head-to-Head Experiments Prompt

In this section, we detail the pairwise evaluation prompts used to assess the synthesized instructions. We prompt DeepSeek-v3 to determine which method-synthesized instruction set is superior in terms of **Quality** and **Coverage**, respectively. The prompts are shown below.

F.1 Quality Evaluation Prompt

To evaluate the overall quality, we focus on correctness, clarity, and reasoning depth. The prompt used for this assessment is presented in Table 14.

F.2 Coverage Evaluation Prompt

To assess how well the generated instructions encompass the source content, we evaluate the scope, density, and completeness of the retrieved information. The prompt used for this assessment is presented in Table 15.

G Downstream Experiment

G.1 Software and Hardware Details

The implementation is built upon the LLaMA-Factory framework (Zheng et al., 2024). To enhance computational efficiency, we integrate FlashAttention-2 (Dao, 2023) to accelerate the attention mechanism and deploy vLLM (Kwon et al., 2023) for high-throughput generation. All experiments were conducted on a computational cluster equipped with NVIDIA A6000 GPUs.

G.2 Hyperparameters

The hyperparameters and technical configurations for the instruction tuning process are documented in Table 8.

Hyperparameters	**Values**
Q-LoRA rank (r)	64
Q-LoRA scaling factor (α)	4
Q-LoRA dropout	0
Optimizer	Paged AdamW
Learning rate scheduler	linear
learning rate	$1e-04$
Weight decay	0
Dropout	0
gradient norm	0.3
Effective batch size	16
Max. input length	2,048
Max. output length	2,048

Table 8: The hyperparameters used for training.

H Use of AI Assistants

In this study, we utilized AI-powered tools, including ChatGPT and Gemini 3, to enhance the linguistic accuracy of our manuscript through spell-checking and minor grammatical corrections. Additionally, the codebase was developed using Claude Code to improve coding efficiency and accuracy.

Filter Prompt for Candidate Text Combination Evaluation

System Instruction
You are a strict evidence-based evaluator for candidate text combinations used in multi-hop question generation. Judge only against the given entity-relation triples.
User Prompt Template
Entity Relations (evidence): `{entity_relations}`
Candidate Text Combinations (to be scored): `{candidate_combinations}`
Each candidate is a pair of related texts; score each pair on two integer dimensions.
Scoring Rubric — Two Dimensions, Each 1–5
Factual Consistency (`s_fact`): 5 fully supported; 4 mostly supported; 3 partially supported/uncertain; 2 many unsupported claims; 1 contradiction/fabrication.
Logical Complementarity (`s_comp`): 5 clear multi-hop dependency; 4 useful 2-hop bridge; 3 weak topical link; 2 forced link; 1 no meaningful link.
Rule: if a claim lacks matching triples, treat it as unsupported.
Output Format
Return a JSON array, one object per candidate:

```
[
  {
    "candidate_id": 1,
    "factual_score": <1-5>,
    "complementarity_score": <1-5>
  }, ...
]
```

Post-processing: $s_{\text{avg}} = (s_{\text{fact}} + s_{\text{comp}})/2$; keep only if $s_{\text{fact}} \geq 3$ and $s_{\text{comp}} \geq 3$, then rank by s_{avg}.

Table 9: The filter prompt for evaluating candidate text combinations based on factual consistency and logical complementarity.

PromptLLM_score for Subgraph Expansion

System Instruction
You expand a KG subgraph for multi-hop question generation by selecting the best candidate edges.
User Prompt Template
Existing Subgraph: `{subgraph}`
Starting Entity: `{start_entity}`
Candidate Edges (connected to the starting entity): `{candidate_edges}`
Task
Select up to `{b}` (max 10) edges and score each candidate in $[0, 1]$.
Selection Criteria (in priority order)
1) **Information Gain**: prefer novel facts/entities; penalize redundancy.
2) **Reasoning Value**: prefer edges enabling 2–3 hop chains.
3) **Core Connectivity**: prefer direct, semantically meaningful links.
4) **Diversity**: avoid duplicated relation/entity patterns.
Output Format
Return JSON only:

```
{
  "reasoning": "brief rationale",
  "selected": [<candidate_numbers>],
  "scores": {
    "<candidate_number>": <score>,
    ...
  }
}
```

Table 10: The PromptLLM_score prompt for selecting candidate edges to expand a knowledge graph subgraph.

Example A: Yes-or-No Question Generation

Instruction
Generate three questions from the source text based on the provided rules and examples. Your generated questions must reflect the logic of the source text and conform to the Yes-or-No Question Answering task format.

Rules
- **Task Type**: Yes-or-No Question Answering.
- **Logical Consistency**: The question must be logically inferable from the input source.
- **Output Format**: The answer must conclude with (but is not limited to) Yes, No, or Maybe.
- **Context Usage**: When multiple sentences are provided, combine them to form a meaningful inference.
- **Style**: The phrasing should be clear, formal, and grammatically correct.

Few-shot Exemplars
Input: *The water in the lake froze overnight due to the sub-zero temperatures.*
Output: Did the temperature drop below zero? (Answer: Yes)

Input: *John looked for his keys in the car but could not find them.*
Output: Did John successfully locate his keys inside the vehicle? (Answer: No)

Selected Context
Input: *The new policy requires all employees to wear ID badges, but visitors are exempt.*

Response
Output: Are visitors required to wear ID badges? (Answer: No)

Table 11: An example prompt instantiated for the Yes-or-No Question Generation task.

Example B: NLI Pair Generation

Instruction
Generate one Natural Language Inference (NLI) pair (Hypothesis + Label) from the source text based on the provided rules and examples. The pair must be logically consistent with the source text and conform to the NLI task format.

Rules
- **Task Type**: Natural Language Inference (Premise $\rightarrow$ Hypothesis).
- **Logical Consistency**: The Hypothesis must be evaluated against the Source (Premise).
- **Label Space**: Must be one of *Entailment* (True), *Contradiction* (False), or *Neutral* (Unknown).
- **Context Usage**: Combine information from the source to form a coherent Premise.
- **Style**: The Hypothesis should be clear and distinct from the source verbatim.

Few-shot Exemplars
Input: *A soccer player is sprinting across the green field chasing the ball.*
Output: Hypothesis: A person is playing a sport. (Label: Entailment)

Input: *The man is sleeping soundly on his couch.*
Output: Hypothesis: The man is running a marathon. (Label: Contradiction)

Selected Context
Input: *Two women are having a conversation near the park entrance.*

Response
Output: Hypothesis: The women are discussing politics. (Label: Neutral)

Table 12: An example prompt instantiated for the Natural Language Inference (NLI) task.

CoT Prompt for Enhanced KG Reasoning

System Instruction
You are a faithful reasoner over an Enhanced Knowledge Graph (Enhanced KG). In an Enhanced KG, each edge carries three elements: entities (head/tail), a relation, and associated context (documents linked to that edge). You must leverage all three elements — entity, relation, and context — at every reasoning step. Use only the provided Enhanced KG subgraph. Do not introduce external knowledge.

User Prompt Template
Question: `{Q}`
Enhanced KG Subgraph: `{G_sub}`

Each edge has the form: `(head, relation, tail, context)`
- **head/tail**: entities (graph nodes)
- **relation**: the semantic relationship between entities
- **context**: documents or text snippets attached to this edge, providing fine-grained evidence

Task: CoT-guided Answer Generation
Reason over the Enhanced KG by jointly using entities, relations, and their associated contexts. Follow the three stages below.

Stage 1: Entity Anchoring
Identify key entities in the question and locate their corresponding nodes in the subgraph.

Stage 2: Reasoning Path Construction
Traverse the Enhanced KG step by step. At each hop, reason over all three elements.
For each hop, output:
```
Step k: (head) -[relation]-> (tail)
  Context: "<relevant information from the associated context of this edge>"
  Reasoning: <how this entity-relation-context triple advances toward the answer>
```
Requirements:
- Each hop must follow an actual edge in the provided subgraph.
- Extract the key information from each edge's context that is relevant to the question; do not merely cite the context, explain what it contributes.
- When consecutive hops involve different contexts, explicitly bridge them: state what information carries over and how the new context extends it.
- If contexts from different edges conflict, flag the inconsistency and prefer the context directly attached to the more specific relation.

Stage 3: Joint Reasoning & Answer
Synthesize the full reasoning path — the chain of `(entity, relation, context)` triples — into a coherent answer.
- The answer must integrate both the structural information (entity-relation path) and the semantic details (contexts along the path).
- Every claim in the answer must be traceable to a specific step above.
- If the subgraph is insufficient to fully answer the question, state what can be answered and what remains unsupported.

Output Format
```
### Entity Anchoring
- "mention_in_question" -> node_in_graph
- ...
### Reasoning Path
Step 1: (head) -[relation]-> (tail)
  Context: "key information extracted from this edge's context"
  Reasoning: how this advances the answer
Step 2: (head) -[relation]-> (tail)
  Context: "key information extracted from this edge's context"
  Bridge: Step 1 established X; this edge's context further reveals Y.
  Reasoning: how this advances the answer
...
### Final Answer
<answer with inline references to steps, e.g., (Step 1), (Step 2)>
```

Table 13: The CoT prompt for answer generation over the Enhanced KG subgraph.

Instruction Quality Evaluation Prompt

System Instruction
You are an expert linguistic evaluator and a helpful assistant. Your task is to compare the quality of two sets of instructions generated from the same source text. You will act as an impartial judge.

Evaluation Criteria (Quality Focus)
Please assess the "Quality" based on the following dimensions:
1. **Fluency & Grammaticality**: Are the instructions written in correct, natural-sounding language?
2. **Clarity**: Is the intent of the instruction clear and unambiguous?
3. **Coherence**: Do the instructions make logical sense?
4. **Appropriateness**: Are the instructions relevant without hallucinating information?
5. **Complexity**: Do the instructions represent a meaningful task?

Constraint: Force a Decision
You usually must choose a winner. Ties are strictly prohibited. If the outputs are very similar in quality, you must identify subtle differences (e.g., slight improvements in conciseness, tone, or structural logic) to determine which one is marginally better.

Input Data
Source Text: `{{SOURCE_TEXT}}`
Model A Instructions: `{{MODEL_A_OUTPUT}}`
Model B Instructions: `{{MODEL_B_OUTPUT}}`

Output Format
First, provide a brief explanation of your reasoning. Then, conclude with your verdict strictly in the following JSON format:
`{ "reasoning": "...", "winner": "Model A" | "Model B" }`

Table 14: The pairwise evaluation prompt for assessing instruction quality.

Information Coverage Evaluation Prompt

System Instruction
You are an expert content evaluator and a helpful assistant. Your task is to compare two sets of instructions generated from the same source text based specifically on "Information Coverage".

Evaluation Criteria (Coverage Focus)
1. **Scope & Breadth**: Do instructions cover a wide range of information?
2. **Entity & Relation Density**: Do they capture key entities and relationships?
3. **Completeness**: Do they collectively retrieve the majority of significant facts?
4. **Diversity**: Are the instructions distinct, covering different aspects?
Note: A model that asks about more distinct, valid details should win.

Constraint: Force a Decision
You usually must choose a winner. Ties are strictly prohibited. If both models cover the text well, choose the one that captures slightly more detail, or covers the less obvious/marginal information that the other missed.

Input Data
Source Text: `{{SOURCE_TEXT}}`
Model A Instructions: `{{MODEL_A_OUTPUT}}`
Model B Instructions: `{{MODEL_B_OUTPUT}}`

Output Format
Explain which key facts were covered by the winner but missed by the loser. Then, conclude strictly in the following JSON format:
`{ "reasoning": "...", "winner": "Model A" | "Model B" }`

Table 15: The pairwise evaluation prompt for assessing information coverage.