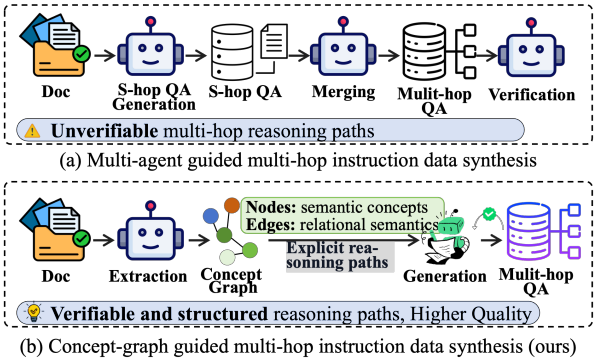
CGMIS: Concept-Graph Based Multi-Hop Instructions Synthesis for Enhancing Long-Context Reasoning

Zechen Sun[1], **Zecheng Tang**[1], **Juntao Li**[1*], **Wenpeng Hu**[2], **Wenliang Chen**[1], **Zhunchen Luo**[2], **Qiaoming Zhu**[1]

[1]Institute of Computer Science and Technology, Soochow University
[2]Information Research Center of Military Science, PLA Academy of Military Science
{zcsunc, zctang}@stu.suda.edu.cn, ljt@suda.edu.cn, wenpeng.hu@pku.edu.cn, wlchen@suda.edu.cn, zhunchenluo@gmail.com, qmzhu@suda.edu.cn

Abstract

High-quality multi-hop instruction data is critical for enhancing the reasoning capabilities of large language models (LLMs) in complex long-context scenarios, e.g., long-form reasoning. Nevertheless, there is currently a notable scarcity of such datasets within the community, and existing data synthesis approaches typically fail to provide explicit modeling of intermediate reasoning steps, resulting in unverifiable and potentially erroneous samples. To mitigate above issue, we design the Concept-Graph based Multi-hop Instructions Synthesis (CGMIS) framework, which constructs long-form reasoning paths via concept graph traversal and automatically generates verifiable multi-hop data. The CGMIS framework not only guarantees the accuracy and verifiability of the synthesized data but also enables the construction of high-quality multi-hop instruction datasets from arbitrary corpora. Experiments show that fine-tuning with CGMIS-generated data achieves state-of-the-art performance across 13 long-context reasoning tasks on various models, using only 10% of the data volume required by existing methods.



(a) Multi-agent guided multi-hop instruction data synthesis

(b) Concept-graph guided multi-hop instruction data synthesis (ours)

Figure 1: Comparison between traditional agent-based data synthesis and our CGMIS framework. While traditional methods generate multi-hop data through implicit, unverifiable agent interactions, CGMIS uses concept-graphs to model reasoning paths explicitly, enabling verifiable and high-quality multi-hop data synthesis.

Introduction

Recent advancements in long-context large language models (LCLMs) have enabled new opportunities for real-world applications, such as multi-document question answering and project level code analysis that require both understanding and reasoning over lengthy inputs (Liu et al. 2024; Wu et al. 2025; Yu et al. 2025; Zhao, Wu, and Xu 2025). Yet, despite possessing extremely long context windows, e.g., even suppress 1 million (Comanici et al. 2025; MiniMax 2025), LCLMs still fall short on long-form reasoning tasks, especially when compared to their performance under short-context settings (Tang et al. 2025; Hengle et al. 2025).

Recent studies show that post-training large language models (LLMs) on multi-hop instruction tuning (IT) data, which involves instruction-response pairs that require integrating information from multiple disjoint sources or reasoning steps, can significantly enhance their ability to capture long-range dependencies and improve long-context reasoning (Chen et al. 2024, 2025b; Fu et al. 2024; Deng et al. 2025; Xu et al. 2025). To enable scalable data generation, recent work has proposed automated frameworks such as NovelHopQA (Gupta et al. 2025) and MING (Chen et al. 2025b), which generate multi-step reasoning paths through rule-based or multi-agent pipelines. However, these methods often encode reasoning paths implicitly in unstructured natural language rationales or agent dialogues, making logical inconsistencies difficult to detect and inference steps hard to trace, resulting in noisy training data that may contribute to the emergence of hallucinated or unsupported responses (Tang et al. 2024; Zhang et al. 2025a).

To address this, we argue that effective multi-hop data synthesis should explicitly represent the reasoning process in a structured and verifiable form. Among various structured representations, graph-based structures can explicitly model entities and relations as nodes and edges, enabling step-by-step traversal of inference chains. Such structures supports traceable reasoning and demonstrates strong capabilities in capturing long-range dependencies and multi-hop relationships within long-context (Li et al. 2024a; Tatarinov et al. 2025; Lei et al. 2025). Constructing reasoning paths over a concept graph offers a transparent means of ensuring logical coherence and traceability, which are critical for generating high-quality, verifiable multi-hop data.

*Corresponding Author

Building on this insight, we propose the Concept-Graph based Multi-hop Instructions Synthesis (CGMIS) framework, which explicitly models multi-hop reasoning as traversable paths over a structured concept graph to support automated, traceable, and scalable IT data generation. The graph structure naturally supports traceability by representing each reasoning step as a traversable edge between semantic concepts, making the full inference chain explicit and auditable. To ensure path validity and logical coherence, CGMIS integrates LLM-based concept extraction with edge confidence scoring and graph refinement, followed by constrained traversal that enforces semantic plausibility. The self-consistency verification step further checks whether the final instruction-response pair aligns with the generated path, helping reduce hallucinations without external annotations. By structuring reasoning as explicit paths over a concept graph, CGMIS enables automatic construction of high-quality multi-hop IT data from diverse textual corpora, significantly reducing reliance on manual verification and overcoming the core limitations of implicit, untraceable generation in prior approaches.

Our contributions are threefold:

- We propose **CGMIS**, a concept graph-based framework for multi-hop instruction synthesis that explicitly models long-form reasoning as traversable paths over semantic concepts, enabling the **automated**, **scalable**, and **verifiable** generation of high-quality instruction data from arbitrary corpora.
- Extensive experiments show that explicitly structured reasoning paths not only enhance the **faithfulness and interpretability** of synthesized data, but also enable models to achieve **state-of-the-art** long-context reasoning performance using only **10%** of the training data required by existing methods, demonstrating remarkable data efficacy in instruction tuning.
- We provide a detailed analysis on factors such as hop count, data scale, and mixing strategies, reconfirming the effectiveness and efficiency of CGMIS and offering valuable insights for future research in multi-hop data synthesis and long-context reasoning area.

Related Work

Long-Context Data Construction

The performance of long-context language models (LCLMs) is increasingly limited by the scarcity of high-quality training data containing meaningful long-range semantic dependencies (Zhang et al. 2025b; Li et al. 2024b). Although architectural improvements, such as enhanced positional encoding (Zhu et al. 2023; Su et al. 2024; Wu et al. 2024) and extended context training (Peng et al. 2023), have expanded the theoretical context window, their practical effectiveness depends on training examples that require integration of information across distant text spans. Current long-context data construction methods mainly rely on corpus-based expansion, such as up-sampling long documents (Yang et al. 2025) or concatenating short contexts (Tworkowski et al. 2024; Song et al. 2025). While these approaches increase input length, they often yield contexts with weak semantic coherence and artificial structure, failing to establish authentic long-range dependencies. More structured strategies use document relevance to link passages (Gao et al. 2024; Li et al. 2025), but still struggle to capture deep relationships among information. As a result, such methods provide limited support for training models to perform complex reasoning when key information is scattered across loosely connected segments.

Multi-Hop Data Synthesis for Long Contexts

Multi-hop reasoning, which involves integrating information dispersed across a long document, is essential for evaluating and training LCLMs (Yen et al. 2025; Kuratov et al. 2024; Fu et al. 2024; Chen et al. 2025a). Existing long-context multi-hop datasets are largely designed for evaluation and lack the scale and diversity needed for effective instruction tuning (Trivedi et al. 2022; Bolotova-Baranova et al. 2023). Rule-based or manually constructed data (Zhu et al. 2024; Gupta et al. 2025) offer high quality but do not scale, limiting their use in large-scale training. Recent work explores automated synthesis through multi-agent frameworks. For example, LongMIT (Chen et al. 2025b) generates multi-hop data by first producing single-hop QA pairs and then linking them via iterative validation, resulting in an inefficient pipeline with little control over reasoning paths. In contrast, our approach builds on conceptual graph structures to explicitly model semantic relations, enabling controlled sampling of valid reasoning paths and the generation of high-quality multi-hop instruction data. The method is generalizable across different domains and supports multi-hop instances with variable hop counts.

Method

In this section, we present the details of Concept-Graph based Multi-hop Instructions Synthesis (CGMIS) framework, which enables automated and traceable generation of multi-hop instruction data. It consists of three core components: Concept-based Graph Construction, Multi-hop Reasoning Path Extraction, and Self-validated Question Synthesis. The workflow of the framework is detailed in Algorithm 1, and the overall architecture is shown in Figure 2.

Concept-based Graph Construction

Long-context reasoning can be improved by representing text as a concept graph, which captures semantic and contextual relationships and supports the identification of multi-hop reasoning paths for instruction tuning. The process begins by splitting the long document into segments, each assigned a unique identifier. This segmentation enables finer-grained analysis and reduces the computational costs on LLMs when extracting conceptual relationships. Next, LLMs extracts concepts and their semantic relations from each segment, assigning an initial weight W_1 to each relation. Additionally, concepts that co-occur within the same segment are linked with a contextual weight W_2. Repeated co-occurrence of concept pairs across segments indicates stronger associations. We then merge duplicate concept

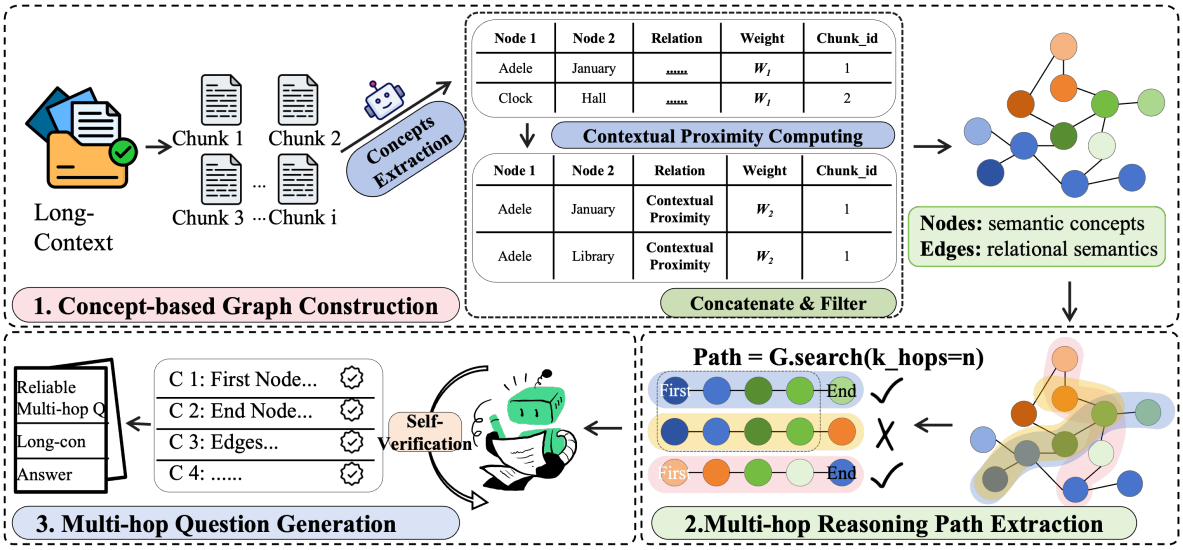


Figure 2: Overview of our Concept-Graph based Multi-hop Instructions Synthesis (CGMIS) framework.

pairs, combining their weights into a single edge. The final edge weight $w(u, v)$ is the sum of all W_1 (semantic) and W_2 (contextual) contributions across segments.

The result is a concept graph $\mathcal{G} = (V, E)$, where V represents unique concepts and E contains weighted edges encoding both semantic and contextual relations. This graph organizes the content and explicitly defines valid reasoning paths for constructing multi-hop QA pairs. By effectively aggregating variations in relationships and weights across all segments, we ensure each concept pair has a single, well-defined edge annotated with a combined weight and a list of associated relation types.

Multi-hop Reasoning Path Extraction

Generating multi-hop questions requires identifying valid reasoning paths in concept graphs, where each hop corresponds to a logical step between connected concepts. This approach enables better control over the quality of the generated data. We perform path search and extraction based on the concept graph constructed in the previous step.

The path search phase identifies subgraphs that match the desired number of hops. For example, to generate a four-hop question, the system finds subgraphs with five nodes and four edges, consistent with the structure of an n-hop path:

$$\mathcal{S}_n = \{(V_s, E_s) \mid |V_s| = n + 1, |E_s| = n\}, \quad (1)$$

where $\mathcal{S}_n$ denotes the set of all subgraphs suitable for n-hop reasoning, with V_s and E_s representing the nodes and edges in the subgraph, respectively.

Once a valid subgraph is identified, the path extraction stage selects a coherent sequence of nodes and edges to form the reasoning path. The method supports paths of arbitrary length and filters out those with repeated concept nodes, promoting diversity in the generated data. The extracted path is then used to guide the construction of multi-hop questions, ensuring that each hop corresponds to a real semantic or contextual connection in the graph. This makes the reasoning process explicit and verifiable, with clear alignment between the number of hops and the actual reasoning depth, and maintaining high data quality.

Multi-hop Question Generation

After extracting the reasoning paths, we use LLMs to generate multi-hop questions aligned with these paths. The end node in a path serves as the answer, while the edges leading to it provide the contextual information needed to formulate the question. Such multi-hop data implicitly concatenates distributed information within the context, thereby encouraging models to capture long-range dependencies and improve its reasoning abilities in long-context.

To guide LLMs in generating well-structured multi-hop questions, we design a self-verification mechanism that guides LLMs during question generation through four constraints: First, the question must be formulated using only the context provided by the start node and connecting edges. Second, it should not mention any intermediate or end nodes. Third, the correct answer must require following the complete reasoning path through the edge contexts. Fourth, the question must involve logical inference rather than simple fact retrieval. These principles ensure that the generated questions are grounded in the concept graph, follow the reliable reasoning path, and cannot be solved through shortcuts. As a result, the model adheres to the predefined structure, reduces reliance on manual annotation, and produces high-quality multi-hop instruction data.

Experiments

In this section, we describe the experiments conducted to evaluate the effectiveness of our proposed approach.

Algorithm 1: Concept-Graph based Multi-hop Instructions Synthesis

Input: Long-Context Collection $\mathcal{D}$, Semantic Weight $\mathcal{W}_1$, Contextual Weight $\mathcal{W}_2$, Max Hops h_{max}, Path Weight threshold θ_{path}, Generative Model $\mathcal{M}$.
Output: Validated QA Pairs $\mathcal{Q}$, $\mathcal{Q}_n = (Q_n, A_n)$.
Procedure

Stage 1: Build concept-based graph.
Split $\mathcal{D}$ into chunks $C_1, ..., C_m$.
Initialize graph $\mathcal{G} = (V, E)$, where $V = \emptyset, E = \emptyset$.
for each chunk $C_i \in \mathcal{D}$: **do**
    $V_i, E_i \leftarrow \text{LLM\_Extract}(C_i)$.
    $E \leftarrow E \cup \{(u, v, r, W_1) \mid (u, v, r) \in E_i\}$.
    for each co-occurring $(u, v) \in C_i$: **do**
        $E \leftarrow E \cup (u, v, \text{co-occurrence}, W_2)$
    end for
    Merge edges and prune $E \leftarrow \{e \in E \mid W(e) \geq \theta_{\text{prune}}\}$.
end for
Stage 2: Extract reasoning paths
for $h = 2$ to h_{max}: **do**
    $\mathcal{P}_h \leftarrow \text{GraphTraversal}(\mathcal{G}, h)$
    $\mathcal{P}_h^{\text{valid}} \leftarrow \{P \in \mathcal{P}_h \mid \text{PathScore}(P) \geq \theta_{\text{path}}\}$
end for
Stage 3: Generate & validate QA
$\mathcal{Q} \leftarrow \emptyset$
for each $P \in \bigcup \mathcal{P}_h^{\text{valid}}$: **do**
    $Q, A \leftarrow \text{LLM\_Generate}(P)$
    if $\text{Validate}(Q, A, P)$: **then**
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(Q, A)\}$
    end if
end for
return $\mathcal{Q}$

Dataset Construction

To construct multi-hop instruction tuning (IT) data using CGMIS, we randomly sample 1,000 long-context segments from the training set of BookSum (Kryściński et al. 2022) to avoid data leakage. These contexts are predominantly drawn from book narratives that exhibit inherent long-range dependencies, making them well-suited for building concept-based graphs. In the pipeline, we use GPT-3.5-turbo (OpenAI 2023) to extract core concepts from the contexts. We conduct a comprehensive comparison of both closed-source LLMs (GPT-4o-mini and GPT-4-1106-preview) and open-source LLMs (Qwen-2.5-instruct-72B (Yang et al. 2024) and Qwen-2.5-instruct-1.5B) as generators, evaluating their performance across response time, costs, case studies, and human-rated scores. The experiments show that GPT-4o-mini achieves the best trade-off between generation quality and cost. Therefore, we adopt GPT-4o-mini as the generator for multi-hop question synthesis. Since a single long-context can support multiple valid multi-hop reasoning paths, the 1,000 input contexts result in approximately 3,000 generated 4-hop questions. Each question undergoes a self-verification step to ensure logical coherence and factual consistency.

Experimental Settings

Backbone Models and Baselines We conduct experiments on the popular and representative LLMs with different context window and parameter scales, including the short-context language model **LLaMA-3-8B-Instruct** (Dubey et al. 2024) and long-context language models with 128K context window: **LLaMA-3.1-8B-Instruct**, **Qwen-2.5-7B-Instruct** (Yang et al. 2024), and **Qwen-2.5-14B-Instruct**, enabling a comprehensive evaluation across diverse architectural and capacity settings. For comparison, we benchmark against three prominent long-context instruction-tuning datasets: **LongAlpaca** (Chen et al. 2023) combines 12,000 scientific paper QA pairs sourced from arXiv and PubMed with additional instruction-following examples. **LongAlign** (Bai et al. 2024) comprises 10,000 long-form instruction samples with sequence lengths ranging from 8K to 64K tokens. **LongMIT** (Chen et al. 2025b) features multi-agent-generated QA pairs enriched with multi-hop reasoning over extended contexts. We selected only 18,000 English samples from LongMIT dataset that match the sequence lengths of the other datasets.

Training Details In our training framework, we organize QA pairs for each long-context into multi-turn conversations, then instruction-tune the LLMs on these complete conversations. The maximum input length is set to 81,920 tokens for LLaMA-3.1 and Qwen-2.5 to preserve comprehensive context without truncation. All models are initialized with official weights for fair comparison and use QLoRA (Dettmers et al. 2023) to improving training efficiency while preserving the inherent capabilities of the LLMs. Notably, we achieve remarkable efficiency through Unsloth (Daniel Han and team 2023) kernel optimizations, using CGMIS data to completely train the model for 1 epoch in just 2 hours on a single NVIDIA A800 (80GB) GPU without requiring complex parallelization strategies.

Evaluation Details We adopt the LongBench (Bai et al. 2025) English test suite to conduct evaluation experiments. It is a comprehensive benchmark comprising 13 datasets across 5 diverse tasks in real scenarios: Multi-Document, Summarization, Few-shot, Synthetic, and Code. Given that 95% of contexts in LongBench are under 32K tokens, we set the default context length to 32K, and set 8K only for the original LLaMA-3. The evaluation metrics are consistent with the official LongBench benchmark.

Main Results

The main results of various models trained with different data are shown in Table 1, which show that CGMIS achieves competitive or superior performance across all models despite using only 10% of the training samples used by the baselines. The detailed observations are as follows:

High Data Efficiency and Strong Performance By leveraging data generated by CGMIS, all models achieve an average state-of-the-art performance using only 10% of the data volume required by other datasets. For instance, on LLaMA-3, CGMIS achieves the highest average score (48.1 vs. 46.6 for LongAlpaca and 42.9 for LongAlign), excelling in Code tasks (+9.2 for LongAlpaca). LLaMA-3.1 with CGMIS surpasses LongMIT by 7.6 points, and Qwen-7B with CGMIS outperforms LongMIT by 8.2 points. This

Model	Multi-Document QA				Summarization				Few-shot Learning				Synthetic			Code			Avg.
	HQA	2WQA	Mus	**A**	G-Rp	QMS	M-N	**A**	TREC	Triv	SAM	**A**	P-C	P-R	**A**	LCC	Repo	**A**	
Results on Short-Context Language Models with 8K Context Window																			
LLaMA-3-8B-Instruct	48.2	35.1	24.8	36.0	31.0	22.6	25.4	26.3	71.0	**89.9**	40.6	67.1	3.0	72.5	37.8	55.6	47.0	51.3	43.7
+ LongAlpaca (12K)	**53.5**	**44.9**	**26.3**	**41.6**	31.3	**25.5**	26.2	27.7	71.0	87.6	**40.8**	66.5	12.0	86.0	49.0	49.6	46.5	48.1	46.6
+ LongAlign (10K)	44.3	34.7	22.2	33.7	**32.6**	22.6	**26.4**	27.2	63.3	82.9	32.9	59.7	8.7	81.0	44.8	54.0	44.4	49.2	42.9
+ LongMIT-en (18K)	47.7	35.9	24.1	35.9	31.9	23.4	25.3	26.9	72.0	89.0	39.9	67.0	11.3	**87.3**	**49.3**	60.3	52.2	56.2	47.1
+ **CGMIS (1K)**	52.3	41.7	26.1	40.1	31.7	25.0	24.2	**27.0**	**73.0**	89.6	40.5	**67.7**	**12.7**	84.3	48.5	**61.1**	**53.4**	**57.3**	**48.1**
Results on Long-Context Language Models with 128K Context Window																			
LLaMA-3.1-8B-Instruct	**61.9**	**47.8**	**32.6**	**47.4**	33.6	25.3	25.8	28.2	71.3	92.0	42.2	68.5	17.5	**99.7**	58.6	64.5	49.9	57.2	52.0
+ LongAlpaca (12K)	59.7	45.8	30.2	45.2	30.9	**26.9**	26.2	28.0	72.0	89.4	42.3	67.9	15.3	97.7	56.5	52.6	52.6	57.0	50.9
+ LongAlign (10K)	40.3	33.1	22.2	31.9	31.1	24.1	**26.8**	27.5	**72.7**	87.9	42.3	67.6	10.9	88.8	49.9	52.3	52.5	52.5	45.9
+ LongMIT-en (18K)	51.6	41.1	28.7	40.4	31.1	24.5	25.0	26.9	71.3	84.4	40.0	65.2	3.3	92.7	48.0	41.5	47.0	47.0	45.5
+ **CGMIS (1K)**	59.9	47.0	31.2	46.0	**35.1**	25.0	26.2	**28.8**	71.3	**92.5**	**43.3**	**69.0**	**21.3**	**99.7**	**60.5**	**66.6**	**55.8**	**61.2**	**53.1**
Qwen-2.5-7B-Instruct	57.7	**43.9**	30.8	44.2	18.1	18.3	13.8	16.7	69.0	88.8	31.8	63.2	11.7	**100.0**	**55.8**	17.3	20.4	18.9	39.8
+ LongAlpaca (12K)	51.8	39.7	27.1	39.6	15.9	**18.7**	14.3	16.3	69.7	87.8	31.8	63.1	8.3	69.0	38.7	**51.7**	36.7	**44.2**	40.4
+ LongAlign (10K)	46.2	32.3	27.7	35.4	19.5	18.0	13.6	**17.0**	**73.7**	84.1	31.2	63.0	8.2	70.9	39.5	18.4	28.6	23.5	35.7
+ LongMIT-en (18K)	50.3	35.3	29.9	38.5	15.9	18.1	13.5	15.8	71.0	85.0	23.9	60.0	6.7	72.0	39.3	12.2	19.7	16.0	33.9
+ **CGMIS (1K)**	**60.3**	41.6	**34.4**	**45.4**	**20.3**	15.7	**14.9**	**17.0**	70.3	**89.0**	**31.8**	**63.7**	**11.7**	99.0	55.3	23.1	**35.3**	29.2	**42.1**
Qwen-2.5-14B-Instruct	64.3	54.0	37.4	51.9	29.7	23.8	21.9	25.1	76.7	**88.8**	**44.9**	70.1	15.9	99.5	57.7	34.3	36.2	35.3	48.0
+ LongAlpaca (12K)	63.0	51.6	36.7	50.4	30.0	24.0	24.0	26.0	78.0	85.4	41.8	68.4	9.1	76.8	43.0	58.3	40.8	49.6	47.5
+ LongAlign (10K)	61.3	43.2	31.5	45.3	31.5	23.3	24.3	26.3	77.7	88.6	44.7	70.3	11.3	90.6	51.0	47.9	**48.6**	48.2	48.2
+ LongMIT-en (18K)	62.3	47.0	38.2	49.2	30.9	24.0	24.3	26.4	78.0	87.9	42.3	69.4	10.9	97.7	54.3	45.3	32.6	39.0	47.6
+ **CGMIS (1K)**	**65.9**	**55.6**	**39.3**	**53.6**	**34.9**	**26.0**	**25.1**	**28.7**	**78.7**	88.3	44.6	**70.5**	**19.3**	**99.7**	**59.5**	**60.5**	46.1	**53.3**	**53.1**

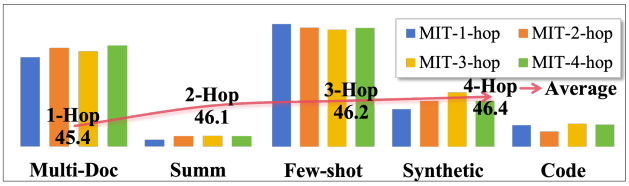
Table 1: Evaluation results on LongBench benchmark, where the **bold** and the underline indicate the best and second-best results, respectively. Both **A** and **Avg.** represent the average score.

demonstrates that our synthetic multi-hop QA data effectively enhances reasoning and information integration capabilities without requiring massive training data. We attribute this to CGMIS generating high-quality, reliable multi-hop data through traceable reasoning paths.

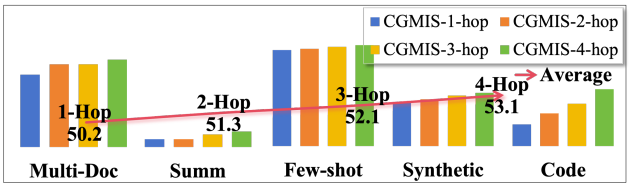
Robustness Across Model Architectures Our CGMIS achieves consistent improvements across all evaluated models, attaining the highest average scores in each case (+4.4 for LLaMA-3, +1.1 for LLaMA-3.1, +2.3 for Qwen-7B, and +5.1 for Qwen-14B). This consistency demonstrates that our multi-hop IT data enhances long-context reasoning capabilities across diverse model families, such as the LLaMA and Qwen series, without requiring architecture-specific tuning, highlighting its generalizability. Notably, the largest gain is observed on Qwen-14B, where CGMIS achieves an average score of 53.1, outperforming LongAlpaca by 5.6 points. This suggests that models with larger parameter capacity can better exploit the complex, multi-hop structure of CGMIS to capture long-range dependencies.

Task-Specific Advantages CGMIS particularly excels in information-intensive tasks requiring cross-context reasoning. For example, on LLaMA-3, our approach improves Synthetic task performance by 10.7 points (48.5 vs. 37.8) and Code tasks by 6.0 points (57.3 vs. 51.3). For the Qwen families, CGMIS demonstrates significant improvements on the Code task, with performance gains of +10.3 on the 7B model and +18.0 on the 14B model. However, other datasets exhibited varying degrees of performance degradation. We claim that traditional IT data maybe reduce LLMs sensitivity to code, while CGMIS-generated multi-hop data emphasizes multi-step logical and structured reasoning, thereby enhancing the LLMs complex reasoning capabilities.

Critical Implications of IT Data for Long-Context LLMs For robust LCLMs like LLaMA-3.1 and Qwen-2.5, traditional IT datasets often degrade performance. For instance, training on LongAlign results in an average performance drop of 6.1 and 4.1 points for LLaMA-3.1 and Qwen-7B, respectively, while LongMIT leads to reductions of 6.5 and 5.9 points. This performance decline maybe related to these datasets reliance on single-hop and 2-hop QA pairs that oversimplify reasoning for advanced models. This suggests that indiscriminate training on generic long-context data can disrupt the inherent capabilities of well-optimized models, particularly in specialized domains like code reasoning. In contrast, our CGMIS employs graph-structured multi-hop IT data (primarily 4-hop) to simulate intricate reasoning paths, aligning better with real-world long-context challenges. While CGMIS slightly underperforms on M-Doc QA tasks for LLaMA-3 variants, this may stem from a domain mismatch: our synthetic data is primarily derived from book-centric multi-hop reasoning, whereas M-Doc QA in LongBench involves a much broader and more varied range of domains (e.g., academic articles and technical reports). Nevertheless, CGMIS achieves significant performance gains across other tasks, validating our primary contribution as a scalable and verifiable method for generating



(a) Performance Comparison of different hops from LongMIT-en



(b) Performance Comparison of different hops from CGMIS

Figure 3: Analysis of the impact of different hop counts based on LongMIT and CGMIS. → shows the **average performance** across the five tasks in LongBench.

high-quality multi-hop IT data that enhances general long-context reasoning capabilities.

Further Analysis

Effects of Hop Counts

We conduct experiments to investigate how instruction tuning data with different hop counts affect reasoning in long-context scenarios. We evaluate LLaMA-3.1-8B-Instruct on two datasets: (1) 1,000 samples per hop (1-4) from LongMIT-en and 1,000 QA pairs per hop created with CGMIS, under the same long-context conditions.

For LongMIT-en data, as shown in Figure 3 (a), shows a slight performance increase with more hops (45.4 to 46.4), especially in multi-document QA (39.1 to 42.3). This suggests a positive correlation between the number of multi-hop training data hops and the model's capabilities in complex reasoning tasks. However, gains are small or negative in some tasks (e.g., 55.34 → 56.33 → 55.34 in Synthetic). This may be due to quality degradation in higher-hop samples from LongMIT-en, which is a known challenge in multi-hop dataset construction, where increasing the number of reasoning steps compromises data credibility. Few-shot performance dips at 4 hops (63.83), hinting that too many hops can add complexity and noise to simpler tasks, supporting our theory that multi-hop data helps more with complex reasoning and there's an **optimal hop-task complexity match**.

CGMIS data (Figure 3 (b)) reveals a consistent positive relationship between hops and performance. Performance increases steadily from 50.2 to 53.1 at 4 hops, with a significant gain of +6.3 on Code tasks. This indicates that CGMIS maintains high data quality even as the number of reasoning steps increases. The results support two key insights: 1) High-quality multi-hop data can enhance model performance in complex long-context reasoning, and 2) Code tasks benefit disproportionately from multi-hop training, suggesting that they inherently require multi-step logical reasoning.

Models	M-QA	Summ	Few-shot	Synth	Code	Avg.
LLaMA-3.1-8B-Instruct						
+ LongMIT	42.3	17.8	63.8	55.3	52.5	46.4
+ **CGMIS-mit**	**42.4**	**27.4**	**66.5**	**58.0**	**59.6**	**50.8**
Qwen-2.5-7B-Instruct						
+ LongMIT	44.0	**18.1**	62.9	54.8	17.1	39.4
+ **CGMIS-mit**	**44.0**	15.6	**64.2**	**55.7**	**23.3**	**40.6**

Table 2: Analysis of different generation frameworks. Evaluation results on LongBench benchmark. CGMIS-mit denotes the reconstructed multi-hop dataset generated using our CGMIS framework, based on the original long-context instances from the LongMIT dataset.

Effects of Mixture Strategies

To explore the impact of mixing multi-hop instruction-tuning data with different hop counts, we conduct experiments with a fixed total of 1,000 samples, selecting data with varying hops and blending them in equal proportions. As shown in Figure 5, 1&4-hop mixture achieves the highest average performance, and the suboptimal result is mixing 2&3&4-hop data in equal proportions, which underscores the necessity of prioritizing high-hop samples in data composition. Notably, 1&2&3&4-hop mixture harms model performance. We attribute this to the over-representation of low-hop data and under-representation of high-hop data, which prevents the model from effectively learning long-range dependencies and may encourage it to exploit shortcuts based on simpler, lower-hop examples. These findings highlight that an imbalanced data distribution, in which **high-hop data is emphasized while a minimal but necessary amount of low-hop data is retained**, is more effective for optimizing long-context reasoning capabilities.

Analysis of Different Generation Frameworks

To eliminate the effect of the initial long-context and to further validate the high quality of CGMIS-generated multi-hop data, we conduct a comparative experiment using 1,000 4-hop QA instances from LongMIT (baseline) and CGMIS-regenerated data (denoted as CGMIS-mit). Both datasets are constructed from identical source documents, with each processed through its respective generation frameworks. As shown in Table 2, models trained on CGMIS-mit achieve average performance improvements of 4.4 for LLaMA-3.1 and 1.2 for Qwen-7B over the LongMIT baselines. This performance gap arises from fundamental methodological differences in data construction. The results indicate that CGMIS framework produces higher-quality multi-hop IT data by concept graphs, thereby more effectively enabling LLMs to capture long-range contextual dependencies. Additionally, we analyze the instances constructed by both frameworks in Figure 4. It is clear that CGMIS-constructed examples strictly follow the reasoning paths defined in the underlying concept graph, while the LongMIT-generated 4-hop data fails to fully instantiate the complex, multi-step reasoning relationships required for multi-hop questions.

LongMIT-4-Hops Question

Which shipping route poses the highest risk for fin whales and how does this relate to the factors affecting feeding success and duration of feeding bouts in whales off the coast of Southern California?

LongMIT-4-Hops Answer

The shipping route that poses the highest risk for fin whales is the Southern route. This information is relevant to the factors affecting feeding success and duration of feeding bouts in whales off the coast of Southern California because the habitat choice, influenced by factors like krill concentration, directly impacts their feeding behavior and success. The study off the coast of Point Mugu and San Miguel Island in Southern California in 2014 and 2015, involving eight blue and five fin whales, revealed differences in feeding intensity and habitat use between males and females of both species. The correlation between feeding bout duration and feeding success suggests that whales remain in areas with higher feeding success longer. However, the presence of shipping routes, particularly the Southern route, which poses a high risk for fin whales, may disrupt these feeding patterns and potentially impact the feeding success and duration of feeding bouts.

(a) Case1: 4-hop case from LongMIT

Concept-based Gragh

Node 1	Node 2	Edge
antibodies	alpha-synuclein	Antibodies have potential for targeting clumping alpha-synuclein proteins.
alpha-synuclein	brain cells	Clumped alpha-synuclein can gum up the inner workings of brain cells.
brain cells	misshapen proteins	The developed nanobody can penetrate brain cells and untangle misshapen proteins.
misshapen proteins	lewy body dementia	Misshapen proteins are also associated with Lewy body dementia.

CGMIS-4-Hops Question

What neurological condition might be potentially alleviated by antibodies that target a protein known to clump and gum up brain cells, which in turn obstructs the normal functioning of those cells by creating misfolded protein tangles?

CGMIS-4-Hops Answer

lewy body dementia

(b) Case2: 4-hop case generated by CGMIS with same long-contexts from LongMIT

Figure 4: Case Study: Comparison between LongMIT case and our CGMIS case with the same long-contexts.

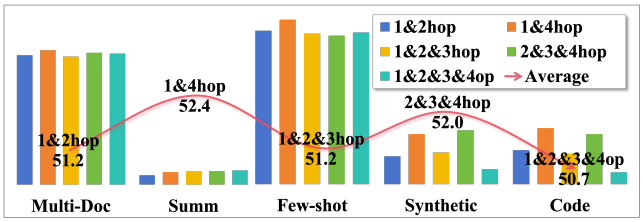


Figure 5: Analysis of the impact of different data mixture strategies based on different hop counts. → shows the **average performance** across the five tasks in LongBench.

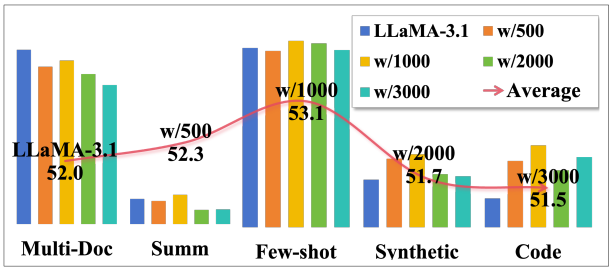


Figure 6: Analysis of the impact of different dataset sizes based on LLaMA-3.1-8B-Instruct. → shows the **average performance** across the five tasks in LongBench.

Effects of Data Size

To investigate the impact of multi-hop IT data scale on model performance, we incrementally expand the CGMIS dataset to 3,000 samples and train LLaMA-3.1. As illustrated in Figure 6, the results reveal a non-linear relationship between data quantity and model performance. Across five long-context tasks, average performance improves steadily as the data scale increases to 1,000 samples (52.0 → 53.1), with code generation exhibiting the most significant gain (57.2 → 61.2). However, further scaling to 2,000 samples leads to performance degradation, ultimately falling below that of the original LLaMA-3.1 (51.7 vs. 52.0). This counter-intuitive inverse trend suggests that moderate amounts of high-quality IT data are sufficient to enhance the complex reasoning capabilities of well-trained LLMs, whereas excessive data scaling may introduce optimization conflicts and overfitting risks. Notably, Multi-Document QA performance worsens with more data. This may be due to the task's similarity to CGMIS and its narrow domain, which could lead to shortcut learning (Geirhos et al. 2020) and hinder cross-domain generalization. It is recommended to design instructions that ensure a delicate balance between logical complexity and domain variety, especially for frontier LCLMs.

Conclusions

In this work, we propose CGMIS, a novel automated framework for generating high-quality, long-context multi-hop instructions through concept graphs. CGMIS consists of three core components: (1) concept graph construction for structured knowledge representation; (2) sampling of multi-hop reasoning paths to capture complex dependencies; and (3) generation and self-verification of multi-hop question-answer pairs, ensuring high data quality, explicit reasoning-path control, and cost efficiency. Notably, CGMIS is domain-agnostic and can be readily applied to diverse corpora, enabling scalable and reliable instruction data synthesis for real-world applications. Extensive experiments show that CGMIS significantly and consistently improves model performance on a range of long-context reasoning tasks, using only 10% of the data required by conventional methods. This highlights the effectiveness, efficiency, and scalability of CGMIS, as well as the critical role of structured knowledge representation in advancing reasoning capabilities in large language models. Furthermore, comprehensive ablation and analysis studies offer valuable insights into optimizing data construction strategies for enhancing long-context reasoning in advanced long-context languages models.

Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. KYCX25_3467), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Bai, Y.; Lv, X.; Zhang, J.; He, Y.; Qi, J.; Hou, L.; Tang, J.; Dong, Y.; and Li, J. 2024. LongAlign: A Recipe for Long Context Alignment of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 1376–1395.

Bai, Y.; Tu, S.; Zhang, J.; Peng, H.; Wang, X.; Lv, X.; Cao, S.; Xu, J.; Hou, L.; Dong, Y.; et al. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3639–3664.

Bolotova-Baranova, V.; Blinov, V.; Filippova, S.; Scholer, F.; and Sanderson, M. 2023. WikiHowQA: A comprehensive benchmark for multi-document non-factoid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5291–5314.

Chen, J.; Wu, J.; Xu, Y.; and Zhang, J. 2025a. LADM: Long-context Training Data Selection with Attention-based Dependency Measurement for LLMs. arXiv:2503.02502.

Chen, L.; Liu, Z.; He, W.; Zheng, Y.; Sun, H.; Li, Y.; Luo, R.; and Yang, M. 2024. Long Context is Not Long at All: A Prospector of Long-Dependency Data for Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8222–8234.

Chen, Y.; Qian, S.; Tang, H.; Lai, X.; Liu, Z.; Han, S.; and Jia, J. 2023. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Chen, Z.; Chen, Q.; Qin, L.; Guo, Q.; Lv, H.; Zou, Y.; Yan, H.; Chen, K.; and Lin, D. 2025b. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 27129–27151.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Daniel Han, M. H.; and team, U. 2023. Unsloth.

Deng, Y.; You, Z.; Xiang, L.; Li, Q.; Yuan, P.; Hong, Z.; Zheng, Y.; Li, W.; Li, R.; Liu, H.; et al. 2025. AlayaDB: The Data Foundation for Efficient and Effective Long-context LLM Inference. In *Companion of the 2025 International Conference on Management of Data*, 364–377.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36: 10088–10115.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fu, Y.; Panda, R.; Niu, X.; Yue, X.; Hajishirzi, H.; Kim, Y.; and Peng, H. 2024. Data engineering for scaling language models to 128K context. In *Proceedings of the 41st International Conference on Machine Learning*, 14125–14134.

Gao, C.; Xing, W.; Fu, Q.; and Hu, S. 2024. Quest: Query-centric Data Synthesis Approach for Long-context Scaling of Large Language Model. In *The Thirteenth International Conference on Learning Representations*.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673.

Gupta, A.; Zhu, K.; Sharma, V.; O'Brien, S.; and Lu, M. 2025. NovelHopQA: Diagnosing Multi-Hop Reasoning Failures in Long Narrative Contexts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 26145–26162.

Hengle, A.; Bajpai, P.; Dan, S.; and Chakraborty, T. 2025. Can LLMs reason over extended multilingual contexts? Towards long-context evaluation beyond retrieval and haystacks. arXiv:2504.12845.

Kryściński, W.; Rajani, N.; Agarwal, D.; Xiong, C.; and Radev, D. 2022. Booksum: A collection of datasets for long-form narrative summarization. In *Findings of the association for computational linguistics: EMNLP 2022*, 6536–6558.

Kuratov, Y.; Bulatov, A.; Anokhin, P.; Rodkin, I.; Sorokin, D.; Sorokin, A.; and Burtsev, M. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37: 106519–106554.

Lei, D.; Li, Y.; Li, S.; Hu, M.; Xu, R.; Archer, K.; Wang, M.; Ching, E.; and Deng, A. 2025. FactCG: Enhancing Fact Checkers with Graph-Based Multi-Hop Data. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5002–5020.

Li, J.; Zhang, X.; Wang, X.; Huang, X.; Dong, L.; Wang, L.; Chen, S.-Q.; Lu, W.; and Wei, F. 2025. WildLong: Synthesizing Realistic Long-Context Instruction Data at Scale. arXiv:2502.16684.

Li, S.; He, Y.; Guo, H.; Bu, X.; Bai, G.; Liu, J.; Liu, J.; Qu, X.; Li, Y.; Ouyang, W.; et al. 2024a. GraphReader: Building Graph-based Agent to Enhance Long-Context Abilities of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 12758–12786.

Li, S.; Yang, C.; Cheng, Z.; Liu, L.; Yu, M.; Yang, Y.; and Lam, W. 2024b. Large Language Models Can Self-Improve in Long-context Reasoning. *arXiv preprint arXiv:2411.08147*.

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

MiniMax. 2025. MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention. arXiv:2506.13585.

OpenAI. 2023. Introducing ChatGPT.

Peng, B.; Quesnelle, J.; Fan, H.; and Shippole, E. 2023. YaRN: Efficient Context Window Extension of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Song, M.; Su, Z.; Qu, X.; Zhou, J.; and Cheng, Y. 2025. PRMBench: A Fine-grained and Challenging Benchmark for Process-Level Reward Models. *arXiv e-prints*, arXiv–2501.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Tang, Z.; Sun, Z.; Li, J.; Zhu, Q.; and Zhang, M. 2025. LOGO—Long cOntext aliGnment via efficient preference Optimization. In *Forty-second International Conference on Machine Learning*.

Tang, Z.; Zhou, K.; Li, J.; Ji, B.; Hou, J.; and Zhang, M. 2024. L-CiteEval: Do Long-Context Models Truly Leverage Context for Responding? *arXiv preprint arXiv:2410.02115*.

Tatarinov, N.; Kannan, V.; Srinivasa, H.; Raj, A.; Anand, H. S.; Singh, V.; Luthra, A.; Lade, R.; Shah, A.; and Chava, S. 2025. KG-QAGen: A Knowledge-Graph-Based Framework for Systematic Question Generation and Long-Context LLM Evaluation. arXiv:2505.12495.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Tworkowski, S.; Staniszewski, K.; Pacek, M.; Wu, Y.; Michalewski, H.; and Miłoś, P. 2024. Focused transformer: Contrastive training for context scaling. *Advances in Neural Information Processing Systems*, 36.

Wu, J.; Gu, G.; Zheng, Y.; Yeung, D.-Y.; and Cohan, A. 2025. Ref-long: Benchmarking the long-context referencing capability of long-context language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 23861–23880.

Wu, W.; Wang, Y.; Fu, Y.; Yue, X.; Zhu, D.; and Li, S. 2024. Long context alignment with short instructions and synthesized positions. *arXiv preprint arXiv:2405.03939*.

Xu, C.; Ping, W.; Xu, P.; Liu, Z.; Wang, B.; Shoeybi, M.; Li, B.; and Catanzaro, B. 2025. From 128K to 4M: Efficient Training of Ultra-Long Context Large Language Models. arXiv:2504.06214.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, C.; Lin, X.; Xu, C.; Jiang, X.; Ma, S.; Liu, A.; Xiong, H.; and Guo, J. 2025. LongFaith: Enhancing Long-Context Reasoning in LLMs with Faithful Synthetic Data. *arXiv preprint arXiv:2502.12583*.

Yen, H.; Gao, T.; Hou, M.; Ding, K.; Fleischer, D.; Izsak, P.; Wasserblat, M.; and Chen, D. 2025. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*.

Yu, Y.; Huang, Y.; Qi, Z.; and Zhou, Z. 2025. Training with "paraphrasing the original text" teaches llm to better retrieve in long-context tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 25751–25759.

Zhang, C.; Zhu, X.; Li, C.; Collier, N.; and Vlachos, A. 2025a. Reinforcement Learning for Better Verbalized Confidence in Long-Form Generation. arXiv:2505.23912.

Zhang, J.; Hou, Z.; Lv, X.; Cao, S.; Hou, Z.; Niu, Y.; Hou, L.; Dong, Y.; Feng, L.; and Li, J. 2025b. Longreward: Improving long-context large language models with ai feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3718–3739.

Zhao, Y.; Wu, H.; and Xu, B. 2025. Leveraging attention to effectively compress prompts for long-context llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26048–26056.

Zhu, A.; Hwang, A.; Dugan, L.; and Callison-Burch, C. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 18–37.

Zhu, D.; Yang, N.; Wang, L.; Song, Y.; Wu, W.; Wei, F.; and Li, S. 2023. PoSE: Efficient Context Window Extension of LLMs via Positional Skip-wise Training. In *The Twelfth International Conference on Learning Representations*.