Self-Taught Agentic Long-Context Understanding

Yufan Zhuang[1,2], Xiaodong Yu[1], Jialian Wu[1], Ximeng Sun[1], Ze Wang[1], Jiang Liu[1], Yusheng Su[1], Jingbo Shang[2], Zicheng Liu[1], Emad Barsoum[1]
[1]AMD, [2]UC San Diego

Abstract

Answering complex, long-context questions remains a major challenge for large language models (LLMs) as it requires effective question clarifications and context retrieval. We propose Agentic Long-Context Understanding (AgenticLU), a framework designed to enhance an LLM's understanding of such queries by integrating targeted self-clarification with contextual grounding within an agentic workflow. At the core of AgenticLU is Chain-of-Clarifications (CoC), where models refine their understanding through self-generated clarification questions and corresponding contextual groundings. By scaling inference as a tree search where each node represents a CoC step, we achieve 97.8% answer recall on NarrativeQA with a search depth of up to three and a branching factor of eight. To amortize the high cost of this search process to training, we leverage the preference pairs for each step obtained by the CoC workflow and perform two-stage model finetuning: (1) supervised finetuning to learn effective decomposition strategies, and (2) direct preference optimization to enhance reasoning quality. This enables AgenticLU models to generate clarifications and retrieve relevant context effectively and efficiently in a single inference pass. Extensive experiments across seven long-context tasks demonstrate that AgenticLU significantly outperforms state-of-the-art prompting methods and specialized long-context LLMs, achieving robust multi-hop reasoning while sustaining consistent performance as context length grows.

1 Introduction

Large language models have achieved notable milestones in natural language processing, demonstrating exceptional performance in tasks such as mathematical reasoning, code generation, and conversational understanding (OpenAI, 2023; DeepSeek-AI, 2025). However, effectively comprehending and utilizing long-context inputs remains a major challenge. Complex queries often require models to retrieve multiple relevant pieces of information from extensive contexts and synthesize them coherently. While recent advancements have extended context windows to 128K and even 2M tokens (Dubey et al., 2024; Touvron et al., 2023; Reid et al., 2024), these models still struggle to fully integrate and reason over large-scale contextual information. Recent studies (Liu et al., 2024; Gao et al., 2024) highlight a fundamental challenge in long-context understanding: the disparity between a model's nominal context size—the theoretical maximum input length—and its effective context window, the portion of the input the model actively utilizes for reasoning. This gap significantly impacts the understanding performance, limiting the model's ability to fully comprehend and integrate long-context information.

We introduce a novel framework AgenticLU to enhance long-context comprehension in LLMs. As illustrated in fig. 1, the core of AgenticLU is **Chain-of-Clarifications** (CoC), a process where models enhance their understanding by generating clarification questions, retrieving relevant information from the long context and answering their own clarification questions based on the gathered evidence. Rather than relying on a direct response, CoC helps models refine their reasoning iteratively, resolving uncertainties along the way. We structure the framework into the following two stages.

CoC Path Construction. To collect reliable CoC understanding path, we structure data collection as a tree search, where each CoC step represents a node. We leverage extended inference time to determine the effective clarification questions to ask and the relevant evidence to retrieve. With a search depth of three and a branching factor of eight, AgenticLU successfully retrieves 97.8% of

Code and data is available at: https://github.com/EvanZhuang/AgenticLU.

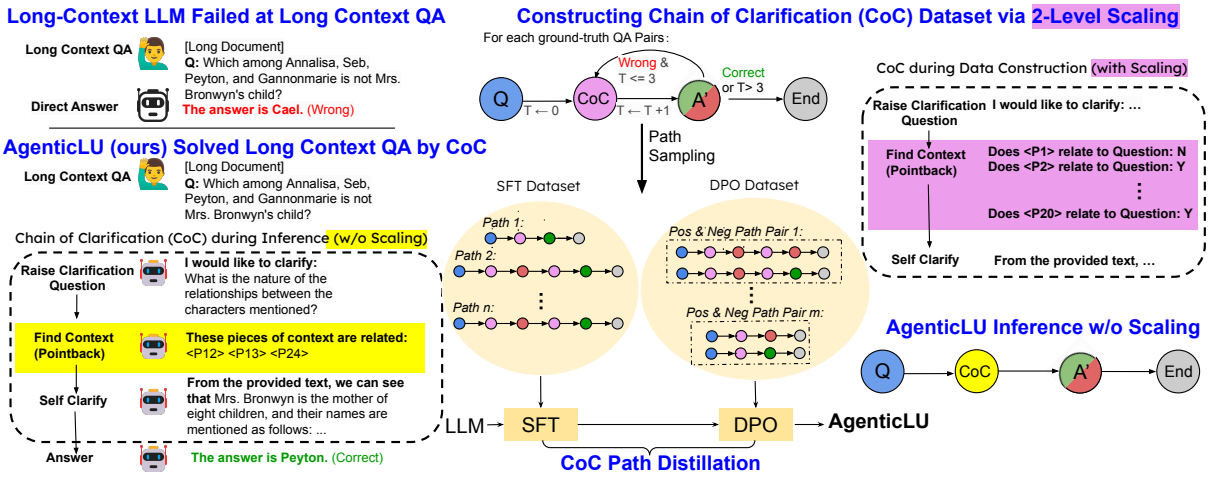

Figure 1: **Overview of the AgenticLU pipeline**: The model iteratively refines its understanding of long-context inputs through an agentic workflow. At each step, it raises self-clarifications, retrieves relevant context via the pointback mechanism, and updates its reasoning trace. The framework integrates CoC Path Construction to generate diverse reasoning paths, followed by two-stage fine-tuning (SFT and DPO) to enhance long-context understanding.

the correct answers in NarrativeQA (Kočiský et al., 2018), demonstrating its capability to tackle complex questions that require multi-step reasoning over long-context inputs.

CoC Path Distillation. Once the dataset is collected from the tree-search process, we train the model to generate effective clarifications and contextual groundings in a single pass, eliminating the need for scaling at inference time. This is achieved by distilling these collected paths into LLMs through supervised finetuning (SFT) and direct preference optimization (DPO) (Rafailov et al., 2024), effectively amortizing the computational cost from inference to training.

Our method AgenticLU significantly improves model's long-context understanding capabilities without relying on laborious human annotations or stronger teacher models for data generation. Instead, the base model's self-generated CoC paths enables it to teach itself to process long-context inputs more effectively. This approach harnesses the model's inherent long-context capabilities—previously only accessible through an additional LLM agent—allowing it to independently refine its reasoning and retrieval processes. Empirically, we demonstrate that AgenticLU consistently boosts performance across a set of question-answering tasks up to 128K tokens, outperforming both prompting-based approaches and other long-context-finetuned LLMs. By integrating self-clarification and context grounding in an agentic manner, we take a step further toward enabling LLMs to comprehend long contexts.

2 Related Work

Challenges in Long Context Understanding LLMs struggle with long contexts despite supporting up to 2M tokens (Dubey et al., 2024; Reid et al., 2024). The "lost-in-the-middle" effect (Liu et al., 2024) and degraded performance on long-range tasks (Li et al., 2023) highlight these issues. To address this, ProLong (Gao et al., 2024) fine-tunes base models on a large, carefully curated long-context corpus. While this approach improves performance on long-range tasks, it comes at a significant cost, requiring training with an additional 40B tokens and long-input sequences.

Inference-time Scaling for Long-Context The Self-Taught Reasoner (STaR) framework (Zelikman et al., 2022) iteratively generates rationales to refine reasoning, with models evaluating answers and finetuning on correct reasoning paths. Wang et al. (2024b) introduced Model-induced Process Supervision (MiPS), automating verifier training by generating multiple completions and assessing accuracy, boosting PaLM 2's performance on math and coding tasks. Li et al. (2024) proposed an inference scaling pipeline for long-context tasks using Bayes Risk-based sampling and fine-tuning, though their evaluation is limited to shorter contexts (10K tokens) compared to ours (128K tokens).

Agentic Workflow for Long-Context Agentic workflows (Yao et al., 2022) enable LLMs to au-

tonomously manage tasks by generating internal plans and refining outputs iteratively. The LongRAG framework (Zhao et al., 2024b) enables an LLM and an RAG module to collaborate on long-context tasks by breaking down the input into smaller segments, processing them individually, and integrating the results to form a coherent output. Chain-of-Agents (CoA) (Zhang et al., 2024b) tackles long-context tasks through decomposition and multi-agent collaboration. In CoA, the input text is divided into segments, each handled by a worker agent that processes its assigned portion and communicates its findings to the next agent in the sequence. Unlike these, our approach employs a single LLM that orchestrates its own reasoning and retrieval without relying on multiple components. By dynamically structuring its process and iteratively refining long-context information, our model reduces complexity while maintaining efficiency.

3 The Context Size Gap

State-of-the-art LLMs have made strong claims about their context lengths, supporting hundreds of thousands of input tokens. However, recent studies (Gao et al., 2024; Yen et al., 2024; Shang et al., 2024) have shown that the *effective* context size of an LLM (the length over which it can reliably perform tasks such as information retrieval and complex reasoning) often diverges from its claimed, or *nominal*, context length.

To illustrate this gap, we evaluate Llama3.1-8B-Instruct, which supports a 128K-token context, on the HotPotQA dataset to test multi-hop QA performance at various input lengths (8K, 16K, 32K, 64K, and 128K). We artificially expand the input by adding irrelevant context and measure the accuracy of its answers using GPT-4o as a judge. As shown in fig. 2, The model's performance degrades substantially as increasing context length, demonstrating the discrepancy between nominal and effective context sizes.

While expanding nominal context capacity is undoubtedly important, we argue that it is not sufficient for solving all long-context problems. By analogy with computer memory, simply having more capacity does not guarantee efficient or accurate computation; one must also manage the "loading" of relevant information in and out of this memory. Therefore, we propose an agentic workflow aimed at helping LLMs process and interpret extended contexts more intelligently.

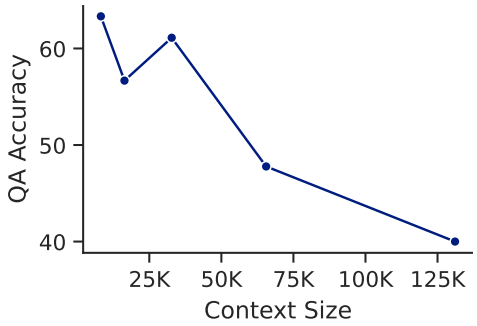

Figure 2: **Effective context size is smaller than nominal context size.** Performance of Llama3.1-8B-Instruct (advertised 128K-token context) on the HotPotQA dataset drops sharply as input length increases (8K, 16K, 32K, 64K, 128K), illustrating the gap between nominal and effective context capacities.

4 Chain-of-Clarifications Workflow

Our approach centers on enhancing long-context comprehension through an iterative, self-refining process that blends inference-time scaling with agentic reasoning. We coin this agentic workflow Chain-of-Clarifications (CoC). In this section, we detail its key components, including the self-clarification process and the pointback mechanism, as illustrated in fig. 1.

Our proposed CoC framework is designed to mitigate the gap between nominal and effective context sizes in large language models. Rather than processing the entire long context and potentially multi-hop questions in a single pass, our methodology decomposes the task into a sequence of targeted sub-tasks. At each CoC step, the model autonomously:

- **Generates clarifying questions** by identifying areas of the long input that require further elaboration or are prone to misinterpretation.
- **Pointbacks to relevant context** by using a pointback mechanism that highlights critical segments of the context by naming the index of relevant paragraphs. In the data collection phase, this is done by iteratively querying the LLM about the relevance of each paragraph with respect to the question. After training, the model is finetuned to generate the related paragraph indexes directly in a single pass.
- **Answers clarifying questions** by integrating highlighted context into consideration to build a more accurate and contextually grounded understanding of the long document.

- **Answers the original question** by combining all newly gathered clarifications, the model attempts to generate a valid answer to the original question.

It is important to note a key distinction between CoC path generation during data collection and the actual task deployment of the agentic workflow. In the data generation phase, we prompt the LLM to iteratively process each chunk of input text along with its self-generated clarifying questions, ensuring accurate retrieval of relevant context. During training, rather than relying on repeated inference calls, we finetune the model to directly generate the indexes of relevant paragraphs using pointback examples, effectively amortizing the computational cost into training. This enables the model to internalize the retrieval process, allowing it to dynamically synthesize relevant clarifications and contextual references at inference time without requiring extensive additional prompting.

5 Data Generation & Model Training

Dataset We use the NarrativeQA (Kočiský et al., 2018) dataset to facilitate long-context QA and generate agentic workflow traces with 14.7K QA pairs in the training set. NarrativeQA is designed for reading comprehension over narrative texts, such as books and movie scripts, where each example includes a full story and a set of corresponding QA pairs. This dataset emphasizes deeper reasoning and long-context understanding, as many questions require synthesizing information from multiple parts of the narrative rather than focusing solely on particular local context. Its relatively long passages make NarrativeQA particularly suitable for testing and refining agentic reasoning in large language models, as the answers often depend on weaving together details spanning the entire text.

Base Model Our base model is *Llama3.1-8B-Instruct* (Dubey et al., 2024), an 8-billion-parameter instruction-tuned Llama model. This model is built on the same transformer architecture as Llama3, but with additional fine-tuning data to improve its performance on multi-turn dialogue and instruction-following tasks.

5.1 CoC Path Construction

We employ a test-time scaling approach to generate CoC paths. For each question, we construct a tree of search paths where each node represents a distinct clarification question posed by the LLM.

Table 1: Statistics of the generated traces dataset used in finetuning derived from NarrativeQA. We left out 11.9K traces for validation.

Data	#
Num of Traces	107,550
Avg Context Length	67,812
Avg Chosen Response Length	165
Avg Rejected Response Length	164
Total Generation Tokens	17M

In our experiments, we use a branching factor of 8 at each depth and select the most promising trace based on an evaluation score that combines:

- **Semantic similarity**, measured by the RougeL (Lin, 2004) score relative to the ground truth.
- **Discrete correctness**, evaluated by a binary verification using GPT4o-mini.

In the data construction process, the relevant context is found by iteratively querying the LLM about the relevance of all chunked passages. Here we use 512 as the chunk size. This process is compute-intensive but only happens in data collection. After the training, the LLM will directly generate the paragraph numbers of the relevant context as shown in the lower right of fig. 1.

For most long-context tasks, a single clarification question suffices because the required reasoning is not highly complex. 92% of the questions in our experiments are resolved correctly with just one round of clarification. More challenging tasks may require multiple rounds of clarification: two rounds resolve 53% of the remaining 8%, and three rounds resolve 35% of the remaining 4%. Because of the exponentially increasing cost—and given that 97.4% of the training questions are already solved—we limit the maximum depth of our inference scaling to 3.

The statistics of the collected dataset are shown in table 1. The total number of conditional generation tokens that the LLM trained on is 17M tokens, with input that has an average length of 67K and a max length of 128K tokens.

5.2 CoC Path Distillation

We employ a two-stage finetuning recipe: Supervised Fine-Tuning (SFT) followed by Direct Preference Optimization (DPO) (Rafailov et al., 2024), to convert our base model into a long-context under-

standing agent. The dataset statistics is described in table 1, with input length up to 128K tokens.

Supervised Fine-Tuning In the first phase, we finetune *Llama3.1-8B-Instruct* using the generated CoC paths. Each training example includes (1) the full context from NarrativeQA, (2) the question, and (3) the step-by-step reasoning trace leading to the final answer. By exposing the model to these traces, we encourage it to internalize multi-step reasoning strategies and context grounding for the long-context inputs. The SFT stage uses a standard cross-entropy loss on the next-token prediction task, ensuring the model learns how to produce consistent and complete reasoning sequences.

Direct Preference Optimization In the second phase, we apply Direct Preference Optimization to further refine the model's output quality. To create preference pairs, we sample incorrect workflow traces as negative examples with using GPT4o-mini as the judge for answer correctness from the test-time scaling. DPO explicitly optimizes the model to generate higher-ranked responses more frequently, thus aligning the agent's outputs with desirable characteristics, such as clarity, correctness, and coherence. This stage ensures that even among valid reasoning paths, the model learns to prioritize the most instructive reasoning.

The details for the two-phase training are listed in appendix A.

6 Evaluation

In this section, we assess our method AgenticLU using a suite of evaluation tasks drawn from the HELMET long-context benchmark (Yen et al., 2024). Our experiments focus on testing models' ability to retain, process, and reason over extended contexts ranging from 8K to 128K tokens.

6.1 Tasks and Metrics

We evaluate our models and baselines on the Helmet (Yen et al., 2024) long-context evaluation benchmark's retrieval-augmented generation (RAG) and long-range QA (LongQA) tasks ranging from 8K, 16K, 32K, 64K, to 128K.

We use GPT-4o as the judge for answer correctness, with the prompt template shown in appendix F. We report accuracies for all datasets.

The RAG test suite includes: (1) **HotpotQA** (Yang et al., 2018), a multi-hop reasoning dataset over Wikipedia; (2) **Natural Questions** (Kwiatkowski et al., 2019), real user queries with Wikipedia-based short and long answers; (3) **TriviaQA** (Joshi et al., 2017), a large-scale trivia dataset with question-answer pairs linked to evidence documents; (4) **PopQA** (Mallen et al., 2022), a dataset testing model memorization with fact-based questions from popular culture.

The LongQA test suite includes: (1) **NarrativeQA** (Kočiský et al., 2018), a reading comprehension dataset with Wikipedia summaries and story-based Q&A; (2) **InfiniteBench QA** (Zhang et al., 2024a), a long-range QA benchmark requiring reasoning over extended contexts; (3) **InfiniteBench Multiple-Choice** (Zhang et al., 2024a), a multiple-choice variant of the previous evaluating reading comprehension over long documents.

For the four RAG tasks, each question is put alongside a set of relevant contexts, and the overall input length is increased by appending irrelevant context. Consequently, these tasks become strictly more difficult as the context window expands. In contrast, for the three LongQA tasks, the relevant context may not appear in the truncated input (the first 8K, 16K, or 128K tokens). Hence, performance might improve at longer input lengths simply because the necessary information becomes available only after including more tokens.

6.2 Baselines

We compare AgenticLU against a diverse set of strong baselines representing different approaches for handling long-context tasks. Our comparisons include two main categories.

Under prompting methods we consider techniques that require no additional model training. In particular, we evaluate (a) the chain-of-thought approach (Kojima et al., 2022), which encourages models to decompose complex questions into intermediate reasoning steps; (b) fact-and-reflection prompting (Zhao et al., 2024c), which iteratively verifies and refines factual claims to enhance consistency; (c) plan-and-solve prompting (Wang et al., 2023), where the model first outlines a high-level plan before sequentially executing it to address structured reasoning tasks; and (d) LongRAG (Zhao et al., 2024a) where a hybrid RAG system is used to retrieve relevant context to generate global summaries and local details [1].

In the fine-tuning category, we focus on models

[1]Note that LongRAG provided finetuned models as well. But the SFT-ed Llama3-8B only supports 8K context length. Thus we did not include it in our comparison.

Table 2: Performance difference of AgenticLU and its base, Llama3.1-8B-Instruct (δ =AgenticLU-8B minus Llama3.1-8B), on long context (the 128K tasks) and short-context benchmarks (6 regular tasks including ARC, GSM8K, and MMLU), the details of the short-context performance can be found in appendix B. Scores represent accuracy, with AgenticLU demonstrating significantly improved performance across long-context tasks with minimal effect on regular task performance.

Model	**Short Avg**	HotpotQA	Natural Questions	TriviaQA	PopQA	NarrativeQA	InfiniQA	InfiniChoice	**Long Avg**
Llama3.1-8B	**62.3**	40.0	56.1	80.6	56.1	38.0	48.0	55.0	**53.4**
AgenticLU (δ)	**-0.6**	+31.1	+21.7	+7.7	+9.4	+18.0	+2.0	+13.0	**+14.7**

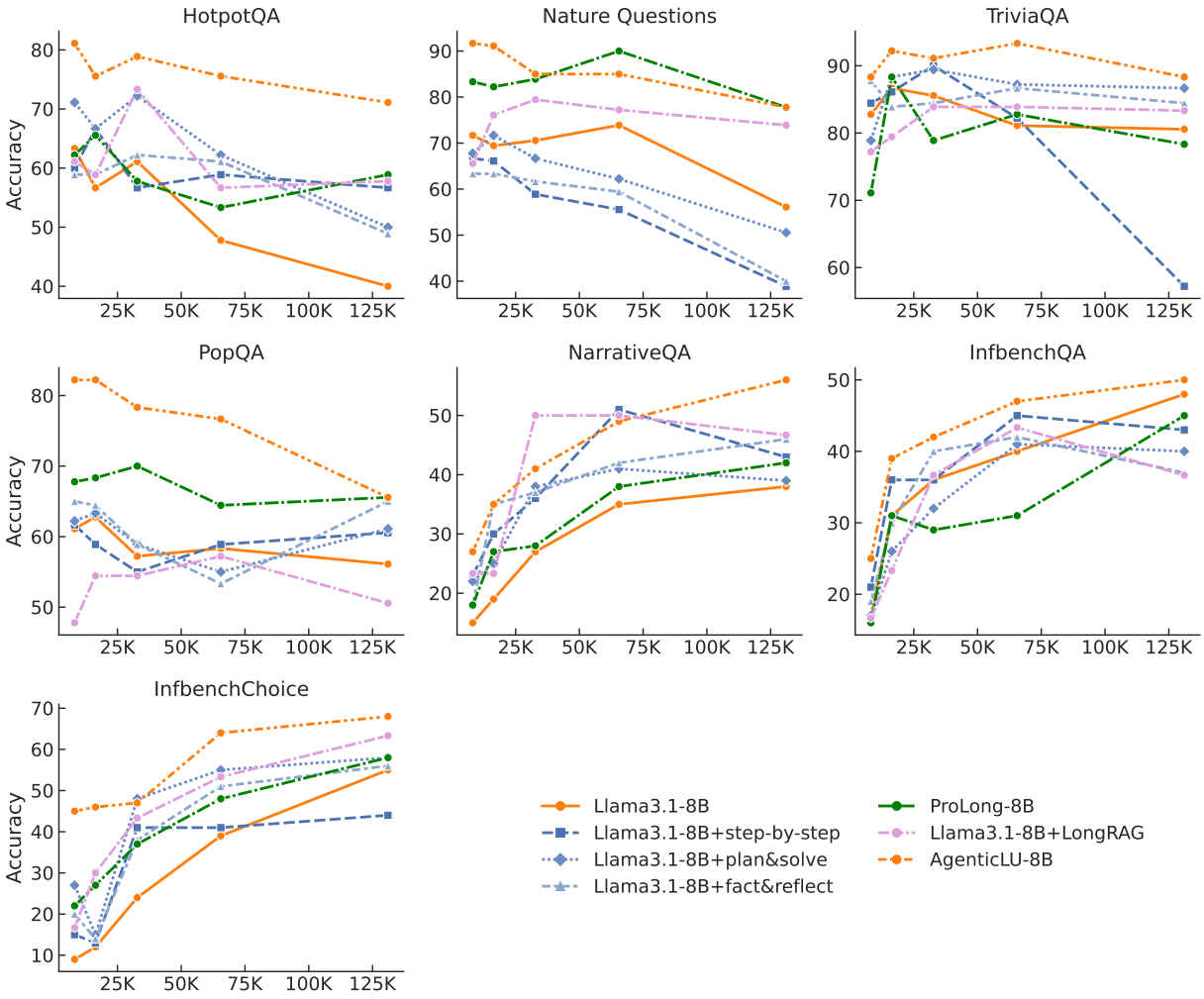

Figure 3: **Main results on 7 long-context tasks across context lengths from 8K to 128K.** Our AgenticLU-8B (dotted orange) achieves significant improvements on *all* tasks over our base model Llama3.1-8B (solid orange). We also compare with the prompting methods (Step-by-Step, Plan-and-Solve, Fact-and-Reflect, LongRAG) and the state-of-the-art ProLong-8B model. AgenticLU-8B consistently maintains strong performance across most tasks and context lengths.

that have been specifically adapted for extended context data. For a substantial comparison, we employ Prolong-8B-512K (Gao et al., 2024)—a model based on the Llama3 8B architecture that has been further trained on an additional 40B tokens of long-context data.

6.3 Main Results

The performance of AgenticLU and baseline models is shown in fig. 3.

Self-clarification significantly improves multi-hop reasoning. AgenticLU-8B consistently surpasses other methods in HotpotQA. By iteratively refining its understanding, resolving ambigu-

ities, and verifying intermediate steps, the model achieves higher accuracy, particularly as context length increases.

Robust performance across diverse datasets. Unlike baseline models, AgenticLU-8B maintains consistently strong performance across RAG and LongQA benchmarks, demonstrating its ability to adapt effectively to different long-context tasks.

Reduced performance degradation with longer contexts. While most models experience significant accuracy drops as context length increases, AgenticLU-8B remains stable. Its self-clarification and pointback mechanisms effectively filter noise from irrelevant information, allowing the model to extract and prioritize essential evidence.

Fine-tuning vs. prompting trade-offs. While structured prompting techniques like *plan-and-solve* improve short-context reasoning, they struggle with extreme context lengths (e.g., 128K tokens). In contrast, AgenticLU-8B, through targeted finetuning with self-clarification and pointback, maintains robust long-context reasoning without relying on complex prompting strategies. Although ProLong-8B, another finetuned model, achieves strong results, it comes with significantly higher training costs. AgenticLU-8B, by contrast, is more data-efficient and generalizes better to novel tasks, making it a more practical and effective solution for long-context reasoning.

Overall, these results underscore the effectiveness of AgenticLU-8B in tackling long-context understanding challenges. The integration of self-clarification plays a crucial role in improving grounding, reasoning, and comprehension in long-context settings.

6.4 Performance on Short-Context Tasks

To demonstrate that our fine-tuning process preserves the model's general capabilities while enhancing long-context understanding, we evaluated the finetuned model on a diverse set of standard benchmarks. These include elementary and advanced reasoning tasks ARC Easy and ARC Challenge (Clark et al., 2018), mathematical problem-solving GSM8K (Cobbe et al., 2021), MathQA (Amini et al., 2019), and broad knowledge assessment MMLU (Hendrycks et al., 2021b,a), MMLU-Pro (Wang et al., 2024a).

We report the average performance across short-context tasks in table 2, and each individual task result can be found in appendix B. We find that the short-context performance is well preserved, demonstrating that AgenticLU's core reasoning and problem-solving abilities remain strong and are not compromised by the significant improvements to its long-context understanding powers.

Table 3: We evaluate the performance of adding additional self-clarification and contextual grounding rounds at inference time. The gain from self-clarification is close to optimal at the initial round.

Model	HotpotQA	NaturalQ	PopQA	TriviaQA	Avg
Llama-3.1-8B	40.0	56.1	56.1	80.6	58.2
AgenticLU-8B	71.1	77.8	65.5	88.3	75.7
(w/ 2 rounds)	71.1	76.7	67.2	91.7	76.7
(w/ 3 rounds)	75.5	78.8	68.3	91.1	78.4

7 Analyses & Ablation Studies

In this section, we take a closer look at how each part of our approach affects long-context understanding and retrieval. Specifically, we study three main questions: (1) Can the finetuned system benefit from multi-round CoC? (2) Does adding clarifications and pointing back to the original document help the model understand and utilize the context more accurately? (3) How much additional compute overhead does AgenticLU add to the process?

7.1 How many rounds of CoC are needed?

Setup. We add additional rounds of reasoning in the evaluation and see if the LLM can benefit from multi-rounds of reasoning at test-time.

Analysis. The results, presented in Table 3, indicate that additional rounds of agentic reasoning do provide performance improvements.

This suggests that while significant benefits of self-clarification are achieved in the first round, additional rounds still contribute to further improvements. One possible explanation is the nature of our dataset: approximately 92% of the questions are resolved within a single round of clarification. However, for the remaining cases, extended reasoning allows the model to refine its understanding, leading to measurable gains in performance with more clarification and reasoning.

7.2 Do Self-Clarifications and Pointback Help in Long-Context Understanding?

Setup. To evaluate the impact of each component in our agentic workflow, we compare the full AgenticLU-8B model against two variants:

Table 4: We test the agentic workflow with AgenticLU-8B when taking out the self-clarification steps and the contextual grounding (pointback) step. The tasks are with 128K context length.

Model	HotpotQA	NaturalQ	PopQA	TriviaQA	Avg
Llama-3.1-8B	40.0	56.1	56.1	80.6	58.2
AgenticLU-8B	71.1	77.8	65.5	88.3	75.7
(w/o Clarification)	57.8	56.7	55.5	78.3	62.1
(w/o Pointback)	53.3	59.4	52.7	83.3	62.2

Table 5: Performance Overhead Comparison between direct answering baseline and AgenticLU.

Metric	Baseline	AgenticLU
Runtime Overhead	100%	101.93%
Avg Tokens Generated in One Round	76.28	1205.38

one without the self-clarification step and another without the contextual grounding (*pointback*) step. We use the four RAG datasets with 128K context length as the evaluation benchmark, and compare the performance alongside the original model.

Analysis. Table 4 shows the results on four QA benchmarks with a 128K context length. Removing self-clarification leads to an absolute performance drop of at least 10 points across most tasks (e.g., from 71.1% to 57.8% on HotpotQA), confirming that the model benefits from clarifying its own uncertainties when the context is long. Meanwhile, omitting pointback yields degenerate results, indicating that pinpointing relevant information at each stage is crucial for long-context QA. Overall, these findings highlight the importance of both clarifications and context-grounding to maximize retrieval accuracy and robustness in lengthy documents.

7.3 How much additional compute cost does AgenticLU impose in generation?

Since additional generation steps are introduced in the QA process, we assess the overhead in inference time. Naïvely, long-context inference and multi-round conversations could significantly amplify compute costs. However, by leveraging prefix caching to store computed KV caches, the additional cost scales linearly with the number of newly generated tokens rather than exponentially.

To quantify this overhead, we conduct a runtime evaluation on 100 queries with a 128K context size. The results, summarized in table 5, demonstrate that the additional computational overhead remains minimal when using prefix caching.

8 Conclusion

In this work, we introduce Agentic Long-Context Understanding (AgenticLU), a framework designed to enhance large language models' ability to process and reason over long-context inputs with self-generated data. By incorporating an agentic workflow (CoC) that dynamically refines model reasoning through self-clarifications and contextual grounding, AgenticLU significantly improves LLM's long context understanding capabilities.

Through a combination of trace data collection and two-stage post-training, our approach enables models to autonomously explore multiple reasoning paths, distill the most effective clarification strategies, and improve their understanding of lengthy documents. Extensive evaluations on long-context benchmarks demonstrate that AgenticLU outperforms existing prompting techniques and finetuned baselines, maintaining strong performance across context lengths up to 128K tokens. Additionally, ablation studies confirm that self-clarification and pointback mechanisms play a crucial role in improving retrieval and reasoning over long-contexts.

Limitations

Despite its effectiveness in long-context reasoning, AgenticLU has notable limitations. One key drawback is its inability to autonomously determine when to stop multi-round reasoning. While additional rounds of self-clarification can improve performance, the model follows a fixed number of reasoning steps rather than dynamically assessing when further refinement is necessary. This can lead to inefficiencies, where the model either stops too early, missing potential improvements, or continues reasoning unnecessarily, expending computational resources without significant gains.

Developing a fully agentic mechanism remains an open challenge. Ideally, the model should assess its confidence in an intermediate response and decide whether further clarification is needed. Future work should explore approaches that enable AgenticLU to regulate its reasoning depth dynamically, optimizing both efficiency and performance.

Acknowledgement

Our work is sponsored in part by NSF CAREER Award 2239440, NSF Proto-OKN Award 2333790, Sponsored Research Projects from companies like Cisco and eBay, as well as generous gifts from

Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

References

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *Preprint*, arXiv:1905.13319.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

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

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jian Hu, Xibin Wu, Weixun Wang, Dehao Zhang, Yu Cao, et al. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*.

Siheng Li, Cheng Yang, Zesen Cheng, Lemao Liu, Mo Yu, Yujiu Yang, and Wai Lam. 2024. Large language models can self-improve in long-context reasoning. *arXiv preprint arXiv:2411.08147*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hao Liu, Matei Zaharia, and Pieter Abbeel. 2023. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*.

OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Jingbo Shang, Zai Zheng, Jiale Wei, Xiang Ying, Felix Tao, and Mindverse Team. 2024. Ai-native memory: A pathway from llms towards agi. *arXiv preprint arXiv:2406.18312*.

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

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024a. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Zihan Wang, Yunxuan Li, Yuexin Wu, Liangchen Luo, Le Hou, Hongkun Yu, and Jingbo Shang. 2024b. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision. *arXiv preprint arXiv:2402.02658*.

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, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024a. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arik. 2024b. Chain of agents: Large language models collaborating on long-context tasks. *arXiv preprint arXiv:2406.02818*.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. 2024a. LongRAG: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22600–22632, Miami, Florida, USA. Association for Computational Linguistics.

Xiaowei Zhao, Yong Zhou, and Xiujuan Xu. 2024b. Dual encoder: Exploiting the potential of syntactic and semantic for aspect sentiment triplet extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5401–5413, Torino, Italia. ELRA and ICCL.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024c. Fact-and-reflection (FaR) improves confidence calibration of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8702–8718, Bangkok, Thailand. Association for Computational Linguistics.

A Training Configurations

We employed the DeepSpeed (Rasley et al., 2020) framework for distributed training across four GPU nodes, each equipped with four AMD MI250 GPUs. We used vLLM (Kwon et al., 2023) for inference. Our implementation builds upon OpenRLHF (Hu et al., 2024) for both SFT and DPO. Given the input sequence length of up to 128K tokens, we leveraged FlashAttention-2 (Dao, 2023) alongside Ring

Attention (Liu et al., 2023) to efficiently process extremely long sequences. The detailed hyperparameters for SFT and DPO are provided in table 6 and table 7.

Table 6: Hyperparameters for SFT.

Hyperparameter	Value
Learning Rate	5e-7
Learning Rate Schedule	Cosine Annealing
Optimizer	Adam
β_1	0.9
β_2	0.95
Training dtype	bf16
Batch Size	128
Max Length	131,072

Table 7: Hyperparameters for DPO.

Hyperparameter	Value
Learning Rate	5e-7
Learning Rate Schedule	Cosine Annealing
Optimizer	Adam
β_1	0.9
β_2	0.95
Training dtype	bf16
Batch Size	128
β	0.1
Max Length	131,072

B Short Context Performance

As shown in table 8, we evaluate the short-context perfromance across six tasks: ARC Easy, ARC Challenge, GSM8K, MathQA, MMLU, and MMLU Pro. AgenticLU performs on par with the base model Llama3.1-8B-Instruct on short-context benchmarks, demonstrating that AgenticLU preserves the original short-context ability while greatly enhancing long-context performance.

C Agentic Workflow without Training

We conducted an additional experiment for prompting only with our agentic workflow, and we find that with an option to generate clarifications the model does get better on multi-hop QA questions (HotPotQA), but it is generally difficult for the base model to point to the correct paragraphs directly without any training, hence often resulting in same or slightly worse performance.

Results are listed in table 9, the context length is 128K tokens.

D Detailed Results on Seven Benchmark Tasks

As shown in table 10, table 11, table 12, table 13, table 14, table 15 and table 16, we evaluate the long-context performance across seven tasks: HotpotQA, Natural Questions, TriviaQA, PopQA, NarrativeQA, InfiniteBench QA and InfiniteBench Multiple-Choice. AgenticLU provides significant improvement for all tasks, especially for those that require multi-hop reasoning such as HotPotQA.

E Chain-of-Clarifications Workflow

The input was first processed into chunks and grouped with paragraph tags. We list the example prompts used in AgenticLU workflow below. In training, we sampled 100 variations of the same prompt text and use them randomly to avoid training collapse.

Chain-of-Clarifications Workflow Prompts

[System Prompt]
You are an AI assistant specialized in long context reasoning. Analyze information thoroughly while maintaining clarity and focus. Track the full context of conversations, building connections between concepts and flagging when context review is needed. Break down complex problems into components, showing your reasoning steps and stating key assumptions. Structure your responses with clear headers and periodic summaries. Present evidence for your conclusions, acknowledge uncertainties, and request clarification when needed. Keep your analysis organized, explicit, and focused on addressing the core question.
[Long-Context Input]
<para 1> [chunk 1] </para 1> <para 2> [chunk 2] </para 2> ... {Question}
[Self Clarification - Raise Question]
In order to answer this question, ask one question about what you want to know in order to better answer it.
[Contextual Grounding - Pointback]
Help me find relevant context to answer the previous clarifying question.
[Self Clarification - Answer Question]
Based on the relevant context, answer the previous clarifying question.
[Answer the Original Question]
Now, let's answer the final question. Be concise in your answer.

Table 8: Performance comparison of AgenticLU and Llama3.1-8B-Instruct on short-context benchmarks. Scores represent accuracy percentages, with AgenticLU demonstrating matching results across tasks.

Model	**ARC Easy**	**ARC Challenge**	**GSM8k**	**MathQA**	**MMLU**	**MMLU Pro**	**Avg**
Llama3.1-8B	84.80	59.64	80.13	42.88	68.72	37.71	62.31
AgenticLU-8B	83.96	58.36	80.51	41.74	68.38	37.51	61.74

Table 9: Performance comparison across different QA benchmarks with 128K token context length, with or without training.

Model	**HotpotQA**	**NaturalQ**	**PopQA**	**TriviaQA**	**Avg**
Llama-3.1-8B	40.0	56.1	56.1	80.6	58.2
Llama-3.1-8B + Prompting Only	53.3	56.7	51.6	72.8	58.6
AgenticLU-8B	71.1	77.8	65.5	88.3	75.7

	HotpotQA				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	63.3	56.7	61.1	47.8	40.0
Llama3.1-8B+step-by-step	60.0	66.7	56.7	58.9	56.7
Llama3.1-8B+plan&solve	71.1	66.7	72.2	62.2	50.0
Llama3.1-8B+fact&reflect	58.9	58.9	62.2	61.1	48.9
ProLong-8B	62.2	65.6	57.8	53.3	58.9
Llama3.1-8B+LongRAG	61.1	58.9	73.3	56.7	57.8
AgenticLU-8B	81.1	75.6	78.9	75.6	71.1

Table 10: Long-context performance on HotpotQA.

	Nature Questions				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	71.7	69.4	70.6	73.9	56.1
Llama3.1-8B+step-by-step	66.7	66.1	58.9	55.6	38.9
Llama3.1-8B+plan&solve	67.8	71.7	66.7	62.2	50.6
Llama3.1-8B+fact&reflect	63.3	63.3	61.7	59.4	40.0
ProLong-8B	83.3	82.2	83.9	90.0	77.8
Llama3.1-8B+LongRAG	65.6	76.1	79.4	77.2	73.9
AgenticLU-8B	91.7	91.1	85.0	85.0	77.8

Table 11: Long-context performance on Nature Questions.

	TriviaQA				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	82.8	86.7	85.6	81.1	80.6
Llama3.1-8B+step-by-step	84.4	86.1	90.0	82.2	57.2
Llama3.1-8B+plan&solve	78.9	88.3	89.4	87.2	86.7
Llama3.1-8B+fact&reflect	87.8	83.9	84.4	86.7	84.4
ProLong-8B	71.1	88.3	78.9	82.8	78.3
Llama3.1-8B+LongRAG	77.2	79.4	83.9	83.9	83.3
AgenticLU-8B	88.3	92.2	91.1	93.3	88.3

Table 12: Long-context performance on TriviaQA.

	PopQA				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	61.1	62.8	57.2	58.3	56.1
Llama3.1-8B+step-by-step	61.7	58.9	55.0	58.9	60.6
Llama3.1-8B+plan&solve	62.2	63.3	58.9	55.0	61.1
Llama3.1-8B+fact&reflect	65.0	64.4	58.9	53.3	65.0
ProLong-8B	67.8	68.3	70.0	64.4	65.6
Llama3.1-8B+LongRAG	47.8	54.4	54.4	57.2	50.6
AgenticLU-8B	82.2	82.2	78.3	76.7	65.6

Table 13: Long-context performance on PopQA.

	NarrativeQA				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	15.0	19.0	27.0	35.0	38.0
Llama3.1-8B+step-by-step	23.0	30.0	36.0	51.0	43.0
Llama3.1-8B+plan&solve	22.0	25.0	38.0	41.0	39.0
Llama3.1-8B+fact&reflect	18.0	35.0	37.0	42.0	46.0
ProLong-8B	18.0	27.0	28.0	38.0	42.0
Llama3.1-8B+LongRAG	23.3	23.3	50.0	50.0	46.7
AgenticLU-8B	27.0	35.0	41.0	49.0	56.0

Table 14: Long-context performance on NarrativeQA.

	InfbenchQA				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	17.0	31.0	36.0	40.0	48.0
Llama3.1-8B+step-by-step	21.0	36.0	36.0	45.0	43.0
Llama3.1-8B+plan&solve	17.0	26.0	32.0	41.0	40.0
Llama3.1-8B+fact&reflect	19.0	30.0	40.0	42.0	37.0
ProLong-8B	16.0	31.0	29.0	31.0	45.0
Llama3.1-8B+LongRAG	16.7	23.3	36.7	43.3	36.7
AgenticLU-8B	25.0	39.0	42.0	47.0	50.0

Table 15: Long-context performance on InfbenchQA.

	InfbenchChoice				
Model	8K	16K	32K	64K	128K
Llama3.1-8B	9.0	12.0	24.0	39.0	55.0
Llama3.1-8B+step-by-step	15.0	13.0	41.0	41.0	44.0
Llama3.1-8B+plan&solve	27.0	15.0	48.0	55.0	58.0
Llama3.1-8B+fact&reflect	20.0	14.0	38.0	51.0	56.0
ProLong-8B	22.0	27.0	37.0	48.0	58.0
Llama3.1-8B+LongRAG	16.7	30.0	43.3	53.3	63.3
AgenticLU-8B	45.0	46.0	47.0	64.0	68.0

Table 16: Long-context performance on InfbenchChoice.

F Evaluation Template

We use GPT-4o (OpenAI, 2023) to judge if the model's answer is correct. The specific prompt template with the structured output class is shown below.

Verification Prompts

```
Please verify the following answer:
Question: question Ground Truth Answers:
ground truth Predicted Answer: answer
Your task is to determine whether the
predicted answer correctly matches the
ground truth. Focus on overall correctness
and provide a detailed explanation in the
following format:
class VerificationResult:
explanation: str # Justification
confidence: float # Confidence score in the
range [0,1]
correct_answer: bool # True if the
prediction is correct, otherwise False
```