AcademicEval: Live Long-Context LLM Benchmark

Haozhen Zhang* *haozhenz@illinois.edu, wazhz14@gmail.com*
University of Illinois at Urbana-Champaign

Tao Feng *taofeng2@illinois.edu*
University of Illinois at Urbana-Champaign

Pengrui Han *phan12@illinois.edu*
University of Illinois at Urbana-Champaign

Jiaxuan You *jiaxuan@illinois.edu*
University of Illinois at Urbana-Champaign

Abstract

Large Language Models (LLMs) have recently achieved remarkable performance in long-context understanding. However, current long-context LLM benchmarks are limited by rigid context length, labor-intensive annotation, and the pressing challenge of label leakage issues during LLM training. Therefore, we propose AcademicEval, a live benchmark for evaluating LLMs over long-context generation tasks. AcademicEval adopts papers on arXiv to introduce several academic writing tasks with long-context inputs, *i.e.*, Title, Abstract, Introduction, and Related Work, which cover a wide range of abstraction levels and require no manual labeling. Moreover, AcademicEval integrates high-quality and expert-curated few-shot demonstrations from a collected co-author graph to enable flexible context length. Especially, AcademicEval features an efficient live evaluation, ensuring no label leakage. We conduct a holistic evaluation on AcademicEval, and the results illustrate that LLMs perform poorly on tasks with hierarchical abstraction levels and tend to struggle with long few-shot demonstrations, highlighting the challenge of our benchmark. Through experimental analysis, we also reveal some insights for enhancing LLMs' long-context modeling capabilities. Code is available at `https://github.com/ulab-uiuc/AcademicEval`.

1 Introduction

Large Language Models (LLMs) have recently achieved tremendous success in natural language processing (NLP) tasks (Achiam et al., 2023; AI@Meta, 2024). However, when facing long context inputs, LLMs show a sharp decline in performance, which poses a pressing challenge to LLMs in understanding and capturing key information in long texts (Li et al., 2024; Liu et al., 2024). Therefore, several long-context LLM benchmarks are spawned to evaluate LLMs in various settings, including question answering, summarizing, and reasoning (Shaham et al., 2023; An et al., 2023; Dong et al., 2023; Bai et al., 2023b; Li et al., 2023; Zhang et al., 2024b). Despite their success, these benchmarks still suffer from concerns of rigid context length, saturated performance, and being leaked in LLM training.

We envision that the *next-generation long-context LLM benchmarks* should ideally possess three key features. (1) *Flexible* and potentially *unlimited* context length: existing benchmarks fix the context for each long-context problem; ideally, the format and length of the context could be flexibly set based on the LLM's capability, especially given the release of long-context LLMs (Reid et al., 2024) and their capabilities in ingesting multi-modal information, *e.g.*, graphs (Dong et al., 2024). (2) High-quality labels derived from *real-world data*, *minimizing* human labeling efforts: existing long-context benchmarks often require human labeling (Bai et al., 2023b; An et al., 2023; Li et al., 2023; Dong et al., 2023; Zhang et al., 2024b), which

*Work done as an intern at University of Illinois at Urbana-Champaign

Table 1: **Comparison with Existing Long-context LLM Benchmarks**. Each column indicates the average input length, whether the annotation is human-assisted, whether there are tasks with hierarchical abstraction levels, whether it contains few-shot demonstrations, and whether the benchmark is lively updated, respectively.

Benchmark	Avg Len	Automatic Annotation	Hierarchical Abstraction	Few-shot Demos	Live Update
ZeroSCROLLS (Shaham et al., 2023)	~10K	✓	✗	✗	✗
L-Eval (An et al., 2023)	~8K	✗	✗	✗	✗
BAMBOO (Dong et al., 2023)	~16K	✗	✗	✗	✗
LongBench (Bai et al., 2023b)	~8K	✗	✗	✓	✗
LooGLE (Li et al., 2023)	~20K	✗	✗	✗	✗
∞Bench (Zhang et al., 2024b)	~200K	✗	✗	✗	✗
AcademicEval (ours)	**Flexible**	✓	✓	✓	✓

is costly and limits the size of the benchmarks to about 2000 samples (Xu et al., 2023) (3) Live updates to mitigate information leakage during LLM pretraining and fine-tuning: benchmark data contamination in LLM has gradually become a severe issue (Sainz et al., 2023; Ye et al., 2024; Zhu et al., 2024b;a; Xu et al., 2024); we argue that holding out future data as the val/test set is one of the most effective approaches for open benchmarks.

Based on these principles, we propose AcademicEval, a live benchmark to evaluate LLMs over long-context generation tasks. AcademicEval adopts arXiv as its data source and features a suite of academic writing tasks on each paper without labor-intensive annotation: Title, Abstract, Introduction, and Related Work, each of which has long-context input and hierarchical abstraction levels. In particular, we construct a co-author graph via the arXiv API to conveniently obtain co-author papers as high-quality and expert-curated few-shot demonstrations, which also possess AcademicEval flexible context length. Furthermore, AcademicEval introduces efficient live evaluation based on the co-author graph, which utilizes the latest papers on arXiv to update the benchmark data periodically and ensures no label leakage. Moreover, AcademicEval provides in-context few-shot demonstrations for each sample, which is neglected by most existing long-context LLM benchmarks (Liu et al., 2024; Li et al., 2024). In our experiments, we evaluate three categories of baselines on AcademicEval: standard LLMs, long-context LLMs, and retrieval-augmented language models (RALM). Under automatic metrics (BERTScore and ROUGE-L), RALM often attains the strongest results by concentrating salient evidence into shorter retrieved chunks, while long-context LLMs and strong standard models remain competitive in several settings. However, an LLM-as-a-Judge evaluation, which assesses novelty, feasibility, consistency, factuality, and academic style, reveals a more nuanced picture: retrieval is not always preferred (*e.g.*, for Title/Abstract), whereas it is highly beneficial for Related Work. Across both evaluations, performance commonly degrades as the input length grows, and correlated few-shot demonstrations from the co-author graph can provide modest gains for specific model–task pairs. Overall, the results indicate that AcademicEval is a challenging benchmark that exposes complementary facets of long-context modeling: overlap-oriented automatic metrics and higher-level judged quality.

We illustrate the comparison with existing long-context LLM benchmarks in Table 1. Our contributions are summarized as follows:

- We propose a live benchmark, AcademicEval, to evaluate LLMs over long-context generation tasks. AcademicEval features four academic writing tasks with hierarchical abstraction levels and requires no manual annotation.

- We construct a co-author graph via the arXiv API and draw on the co-author papers as informative few-shot demonstrations, making the context length of AcademicEval flexible and scalable. Especially, AcademicEval conducts periodic data updates on the co-author graph to enable efficient live evaluation, which ensures no label leakage and fair evaluation.

- We conduct comprehensive experiments on AcademicEval, and the results demonstrate its challenges and yield potential insights for improving LLMs in long-context modeling.

2 Related Work

Long-context Modeling and LLM Benchmarks. LLMs are known to be powerful in language modeling tasks (Achiam et al., 2023; AI@Meta, 2024). However, when it comes to long-context inputs, LLMs show a sharp decline in performance, posing a pressing challenge when benchmarking their long-context modeling capabilities (Liu et al., 2024; Li et al., 2024; 2025). Currently, there are two mainstream technologies for long-context modeling tasks: retrieval-augmented language models (RALM)(Ram et al., 2023; Yu et al., 2023; Trivedi et al., 2022; Jiang et al., 2023; Asai et al., 2023; Zhang et al., 2024a; Feng et al., 2024) and long-context LLMs (Bai et al., 2023a; Jiang et al., 2024; Teknium et al.). RALM equips LLMs with a retriever (Robertson et al., 2009; Ramos et al., 2003; Karpukhin et al., 2020; Izacard et al., 2021) to perform information retrieval on short text chunks, which are then fed to LLMs together with the input query to generate the final output. As a retrieval system, RALM is usually evaluated over retrieval-based benchmarks, including STARK (Wu et al., 2024), RGB (Chen et al., 2024), ARES (Saad-Falcon et al., 2023), etc. In comparison, long-context LLMs expand their context window length to accommodate longer inputs and are benchmarked over various tasks, which include long-context QA, summarization, conversations, reasoning, etc (Shaham et al., 2023; An et al., 2023; Dong et al., 2023; Bai et al., 2023b; Li et al., 2023; Zhang et al., 2024b; Li et al., 2025).

Recent works such as ResearchTown (Yu et al., 2024) and WildLong (Li et al., 2025) share conceptual proximity to our setting but target different goals. ResearchTown is a multi-agent simulation framework that models the dynamics of a research community via message-passing on an agent–data graph, simulating activities such as paper and review writing. Its focus lies in simulating collaborative behavior and ensuring the realism of outputs under controlled settings. In contrast, AcademicEval is a live, real-world benchmark grounded in authentic academic papers, designed to evaluate LLMs on hierarchical writing tasks (Title, Abstract, Introduction, and Related Work) under evolving and leakage-resistant conditions. While both leverage graph structures, ResearchTown uses them for interaction simulation, whereas AcademicEval employs a co-author graph for retrieving high-quality few-shot demonstrations, supporting scalable context lengths, and enabling periodic data updates.

Similarly, WildLong introduces a scalable framework for synthesizing realistic long-context instruction data. It extracts meta-information from user queries, builds co-occurrence graphs, and employs adaptive generation to create 150K instruction–response pairs for complex multi-document reasoning tasks. While WildLong focuses on data synthesis for instruction tuning, AcademicEval focuses on evaluation, providing a live, automatically updated benchmark that measures LLMs' long-context reasoning and generation abilities on real-world academic tasks. Together, these works are complementary: ResearchTown and WildLong contribute to synthetic data generation and simulation, whereas AcademicEval provides a robust evaluation framework for real-world, graph-enabled long-context reasoning.

Label Leakage in LLM Benchmarks. Label leakage has always been a severe issue that benchmarks must attempt to avoid during data collection. However, recent research (Xu et al., 2024; Zhu et al., 2024b;a; Ye et al., 2024) point out that most LLM benchmarks are composed of statically collected data, which may be inevitably included in the large amount of training data of LLMs, causing label leakage. Therefore, some works attempt to measure or detect the extent of label leakage in LLM benchmarks. Benbench (Xu et al., 2024) leverages perplexity and N-gram accuracy to quantify potential label leakage, while PAC (Ye et al., 2024) detects contaminated data by comparing the polarized distance of samples before and after augmentation. Even though these approaches propose to measure or detect label leakage, there is little work on mitigating and solving this issue (Zhu et al., 2024b). Dynabench (Kiela et al., 2021) and Dynaboard (Ma et al., 2021) feature dynamic human-in-the-loop dataset creation while avoiding leakage, which is very labor-intensive. DyVal (Zhu et al., 2024b) leverages pre-set constraints and directed acyclic graphs (DAG) to dynamically generate test cases with diverse complexities, reducing the risk of label leakage. FreshBench (Zhu et al., 2024a) and StackMIA (Ye et al., 2024) collect the latest data from public websites periodically and simply rely on the chronological split to build a dynamic benchmark.

Long-context Summarization Benchmarks. Solving AcademicEval requires LLM's long-context summarization capability (Liu et al., 2024). Existing works include (1) query-based summarization tasks, focusing on the capability of models to position and capture local key information in long texts given a specific query (Litvak & Vanetik, 2017; Wang et al., 2022); (2) single-document or multi-document summarization tasks concentrate on evaluating the ability of models to understand long texts holistically (Cohan et al., 2018; Meng et al., 2021; Huang et al., 2021; Kryściński et al., 2021; Cachola et al., 2020). These long-context summarization benchmarks suffer from the above-mentioned limitations, including requiring human-assisted labeling and concerns about data leakage; moreover, these summarization tasks focus on one-level summarization, failing to consider the summarizations at different abstraction levels.

3 AcademicEval Benchmark

In this section, we propose AcademicEval (Figure 1) for live evaluation over long-context generation tasks with hierarchical abstraction levels. We first describe data collection and preprocessing in Section 3.1. Then, in Section 3.2, four academic writing tasks with diverse abstraction levels are introduced, and we also integrate few-shot demonstrations to make the context length flexible and scalable. Finally, Section 3.3 elucidates the live evaluation with periodic data updates.

3.1 Data Curation

Co-author Graph Construction via arXiv. As a public paper preprint platform, arXiv[1] has always been favored by researchers. It archives a huge amount of papers and updates the latest ones daily, which serves as an excellent data source and also lays the foundation for the live update of our benchmark. Thanks to the arXiv API[2], paper files can be obtained in batch without much manual effort. We first collect and construct a co-author graph (*i.e.*, edges are established between two co-author nodes) using the arXiv API through breadth-first search (BFS), where the features of each author node include the published first-author papers. By making the co-author graph the carrier of papers, we can form an interconnected whole of scattered articles, which provides valuable structural information to be exploited for our benchmark. Furthermore, we can enable efficient live updates on the co-author graph, which will be introduced in Section 3.3.

Academic Data Gathering and Preprocessing. After the co-author graph is collected, we remove authors who have not published independent first-author papers (*i.e.*, appear only as co-authors in the author list) and then prune it to obtain the maximum connected component. For each paper (*i.e.*, node features), we collect essential metadata via the arXiv API, including author information, publication timestamp, etc., and download the PDF file simultaneously, which further goes through a series of pipelines to split and extract the text of several sections in it. In detail, we leverage PyMuPDF[3] to detect section headings (*e.g.*, "Introduction") and extract the paper content by sections. Especially for the "Related Work" section, we extract each cited paper's abstract and title via the arXiv API to form an additional citation corpus. All these processed data constitute the node feature of each author node. We will further describe in Section 3.2 how to use these data to design long-context academic writing tasks.

3.2 Benchmarking LLMs over Long-context Generation Tasks with Hierarchical Abstraction

Task Description. Employing machine learning approaches to automate academic writing has always been a research hotspot with significant practical application value (Chen et al., 2022; 2021). Therefore, inspired by the leave-one-out validation, we introduce four academic writing tasks with ultra-long context to evaluate the generation capability of LLMs under different abstraction levels, as shown below:

[1] https://arxiv.org/

[2] https://info.arxiv.org/help/api/index.html

[3] https://github.com/pymupdf/PyMuPDF

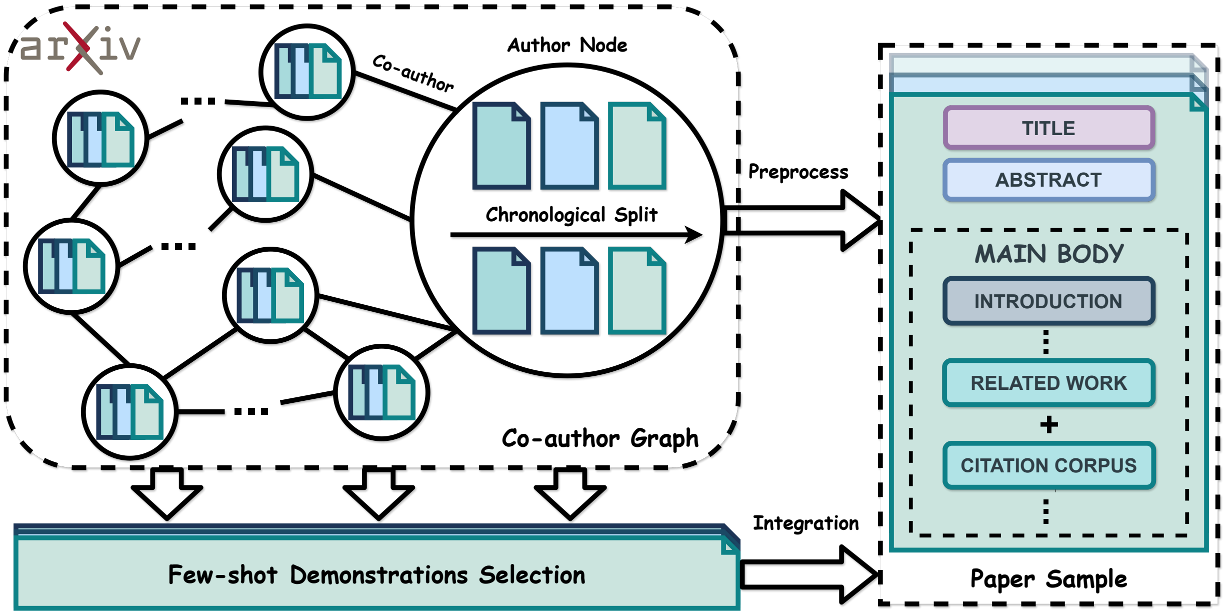

Figure 1: **AcademicEval Benchmark.** We construct a co-author graph via arXiv and conduct a chronological split on all paper samples (training, validation, and test samples are represented by red, orange, and green, respectively). Each paper sample is preprocessed into separate sections and can be integrated with few-shot demonstrations from co-author papers.

- Title Writing. This task takes a paper's main body and abstract, along with a specific task prompt as inputs, and then asks LLMs to output a predicted title.
- Abstract Writing. Similar to the above, this task takes a paper's main body (with the "Conclusion" section removed) and title, along with a specific task prompt as inputs, and then asks LLMs to output a predicted abstract.
- Introduction Writing. This task takes a paper's main body (with the "Introduction" section removed), title, and abstract, along with a specific task prompt as inputs, and then asks LLMs to output a predicted introduction.
- Related Work Writing. This task takes a paper's main body (with the "Related Work" section removed), title, abstract, and citation corpus (introduced in Section 3.1), along with a specific task prompt as inputs, and then asks LLMs to output a predicted related work.

Based on the above task descriptions, we can generate four basic benchmark settings with different abstraction levels, namely Title-10K, Abs-9K, Intro-8K and Related-34K, with suffixes indicating their input context length[4]. Intuitively, the paper content itself can be considered as a kind of original, expert-curated, and high-quality labeled data without manual annotation. Therefore, for evaluation, we directly adopt the corresponding paper section as the ground truth for each benchmark setting, minimizing human labeling efforts.

Integration of Few-shot Demonstrations. Given the rigid context length of current long-context LLM benchmarks and the general effectiveness of in-context learning in LLMs (Dong et al., 2022; Wei et al., 2022a;b; Kojima et al., 2022), we propose to integrate long few-shot demonstrations to enable flexible and scalable context length, and we have two selection options for each sample in the above four basic benchmark

[4] We use BERT (Devlin et al., 2018) tokenizer by default to count the number of input tokens (output tokens are not included).

Table 2: **Data Statistics of AcademicEval (Initial Round).** It includes 4 writing tasks and provides four settings of different context length for each task. For each setting, we list their Comp. Rate, Samples of Each, Chronological Split, and Timespan of Test Data.

Setting	Comp. Rate (In-Len. / Out-Len.)	#Samples of Each.	Chronological Split (Train-Val-Test)	Timespan of Test Data
		Title Writing		
Title-10K	587	5098	72%-19%-9%	2024.06-2024.07
Title-30K	1773			
Title-31K-G	1807			
Title-50K-M	2968			
		Abstract Writing		
Abs-9K	36	5098	72%-19%-9%	2024.06-2024.07
Abs-28K	108			
Abs-29K-G	112			
Abs-48K-M	185			
		Introduction Writing		
Intro-8K	6	4665	71%-20%-9%	2024.06-2024.07
Intro-28K	21			
Intro-28K-G	22			
Intro-48K-M	37			
		Related Work Writing		
Related-34K	34	2240	72%-20%-8%	2024.06-2024.07
Related-53K	53			
Related-53K-G	53			
Related-72K-M	72			

Note: We use the BERT tokenizer by default to count the number of tokens.

settings: *(1) Randomly select papers under the same category.* According to the paper categories provided by the arXiv API, we can randomly select several non-duplicate papers under the same category. *(2) Randomly Select co-author papers.* The motivation is straightforward: the similarity of research directions between co-author papers is more fine-grained. Thanks to the co-author graph, it is convenient to obtain the co-author papers of each original paper sample. These selected papers serve as few-shot demonstrations and are utilized as input-output pairs to enrich the input context of the original samples, providing potentially insightful and relevant content while enabling flexible and scalable context length.

Consequently, we have completed the construction of benchmark settings, and the data statistics in the initial collection round are shown in Table 2.

Data Statistics. As shown in Table 2, AcademicEval has four academic writing tasks with hierarchical abstraction levels, and each task features four settings with diverse input context lengths, some of which are obtained by integrating few-shot demonstrations. For instance, each sample in Title-10K consists of a single paper sample. Title-30K and Title-31K-G are obtained by integrating with two few-shot demonstrations from random papers and co-author papers, respectively, while Title-50K-M is obtained by using both of the above integration options. Actually, we can scale context length by increasing the number of few-shot demonstrations to provide more informative references, enhancing task performance.

Furthermore, we present the text compression rate (defined as the number of input tokens divided by the number of output tokens) for each benchmark setting in Table 2 to illustrate the diverse abstraction levels in AcademicEval. Across the four tasks, a higher compression rate means a higher level of text abstraction

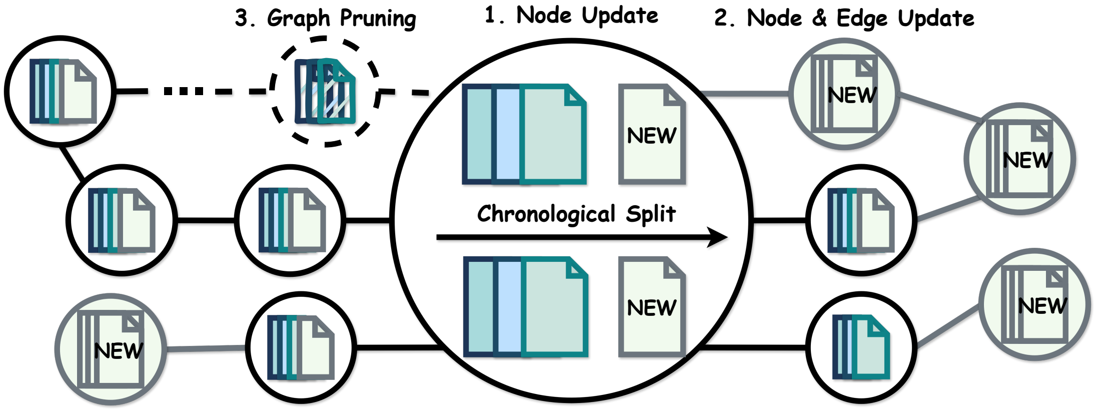

Figure 2: **Live Evaluation of AcademicEval Benchmark.** To support continual benchmarking, AcademicEval incrementally updates the co-author graph using daily arXiv data. The procedure includes: (1) **Node Update** – augmenting node features for authors with newly published first-author papers; (2) **Node and Edge Update** – identifying and prioritizing new co-authors via BFS to expand the graph with recent publications; and (3) **Graph Pruning** – removing outdated papers and inactive authors to maintain graph connectivity and efficiency.

in this task. Among several settings within each task, a higher compression rate makes it tougher to exploit information holistically but more likely to produce better outputs (since more references are integrated). These different tasks and settings increase the diversity of the AcademicEval benchmark.

As for data splitting, we perform a chronological split in AcademicEval, which means that the test set always contains the latest papers collected in each collection round, ensuring no label leakage. Note that Table 2 shows only the data collected in the initial round, which will be updated periodically as described in the next section.

3.3 Live Evaluation with Periodic Data Updates on the Co-author Graph

The daily updates of arXiv provide the basis for the live evaluation of AcademicEval: we can periodically update the benchmark with the latest papers on arXiv. By setting a reasonable update cycle (*e.g.*, monthly or quarterly), we can ensure that the data in the benchmark is not contaminated so that it can be used to evaluate LLMs fairly in a live manner. Therefore, we proposed an efficient incremental update procedure on the co-author graph:

(1) Node Update. For each author on the co-author graph, check whether the author has a newly published first-author paper through the arXiv API. If so, add it to the corresponding node feature on the co-author graph.

(2) Node and Edge Update. During the traversal of Node Update, each author's new co-authors are added to a candidate list, and the number of new papers (including first-author and non-first-author papers) when searching for the author is used as the priority of the co-authors (co-authors of active authors tend to be active as well, and we can efficiently collect the latest papers from active authors). Then, we use the prioritized candidate list to conduct BFS to update nodes and edges until a specific number of incremental update papers is met.

(3) Graph Pruning. As the benchmark is updated, we will remove some outdated papers and inactive authors (defined as those who have not published new first-author or non-first-author papers for a long time) from the co-author graph.

In this way, the latest papers can be obtained sufficiently and efficiently while ensuring connectivity and a smaller graph size.

4 Experiments

4.1 Baselines

We adopt the following three types of baselines to conduct a holistic evaluation of AcademicEval.

Standard LLMs. We choose Gemma Instruct (7B) (Team et al., 2024) and LLaMA-3 Chat (70B) (AI@Meta, 2024) as standard LLM baselines, each with a context length of 8K.

Long-context LLMs. We choose Qwen 1.5 Chat (72B) (Bai et al., 2023a), Mixtral-8x7B Instruct (46.7B) (Jiang et al., 2024), and Nous Hermes 2 - Mixtral 8x7B-DPO (46.7B) (Teknium et al.) as long-context LLM baselines, each with a context length of 32K.

Retrieval-augmented language models (RALM). First, we consider two sparse retrievers: (1) **BM25** (Robertson et al., 2009): This is a widely used retrieval model that ranks documents based on the frequency of query terms in each document. (2) **TF-IDF** (Ramos et al., 2003): It scores documents by multiplying the term frequency of each query term by the inverse document frequency. Second, we also consider three dense retrievers: (3) **DPR** (Karpukhin et al., 2020): It uses a bi-encoder to retrieve relevant documents based on dense embeddings. (4) **Contriever** (Izacard et al., 2021): It leverages unsupervised contrastive learning to learn high-quality dense representations. (5) **Dragon** (Lin et al., 2023): It enhances retriever training by employing data augmentation, including query and label augmentation.

4.2 Settings

API Access. In this paper, we conduct a comprehensive evaluation over AcademicEval benchmark using the LLM API provided by together.ai[5]. For each API call, we fix the temperature parameter to 0 (*i.e.*, greedy decoding).

Input Truncation. By default, we use a BERT tokenizer to calculate the number of input tokens for AcademicEval. However, since the tokenizer of each LLM is usually different, it will cause some inputs to exceed the context length limit of the LLM. Therefore, for the evaluation of each LLM, we additionally download its tokenizer configuration file from the official website at Hugging Face, which is utilized to ensure correct and accurate truncation of input tokens.

Refinement of LLM Responses. For the Title Writing task, the responses of LLMs are relatively short. If the response contains some extra redundant information, it will have a greater impact on the evaluation metric score (although we have given LLM instructions not to generate irrelevant information). Therefore, for the Title Writing task, we additionally refine the LLM responses, for example, removing irrelevant information such as "here is the title". For other tasks, since LLM's responses are relatively long, occasional small amounts of irrelevant information will not have a significant impact on the evaluation, so we do not perform any refinement on LLM's responses in this case.

Details of the Implementation of RALM. We use the inputs of AcademicEval as the external corpus of RALM (such as Target Content and Reference Content introduced in Section D). For text split, we use the RecursiveCharacterTextSplitter from LangChain[6] and set chunk size and chunk overlap to 512 and 64, respectively. For each retrieval, we recall up to 12 text chunks (limited by the context length of standard LLMs) based on text similarity (semantic similarity based on inner product for dense retrievers or similarity based on word frequency for sparse retrievers).

Table 3: **Main Results on AcademicEval w.r.t. BERTScore.**

Models	**Standard LLMs**		**Long-context LLMs**			**RALM**	
	Gemma	LLaMA	Qwen	Mixtral	Hermes	Gemma†	LLaMA†
#Params.	7B	70B	72B	8x7B	8x7B	7B	70B
Context Length	8K	8K	32K	32K	32K	8K	8K
Setting: Title Writing							
Title-10K	66.1	74.1	73.9	73.4	**74.2**	65.8	73.9
Title-30K	-	-	73.0	72.9	73.4	65.7	**73.9**
Title-31K-G	-	-	72.8	72.8	73.3	65.7	**73.8**
Setting: Abstract Writing							
Abs-9K	59.9	62.4	**62.5**	61.4	62.2	60.3	61.5
Abs-28K	-	-	61.3	61.2	**62.6**	60.1	61.4
Abs-29K-G	-	-	61.3	61.4	**62.5**	60.2	61.3
Setting: Introduction Writing							
Intro-8K	54.8	**55.8**	55.4	54.6	55.2	55.0	55.2
Intro-28K	-	-	54.8	54.0	54.8	55.0	**55.2**
Intro-28K-G	-	-	54.9	54.1	54.7	55.0	**55.3**
Setting: Related Work Writing							
Related-34K	52.0	56.2	**58.5**	55.3	57.8	52.4	54.7
Related-53K	-	-	-	-	-	52.4	**54.7**
Related-53K-G	-	-	-	-	-	52.4	**54.8**

Bold indicates the highest score in each row.
† denotes augmentation with a retriever (Default: Contriever).
"-" means that the context length is too long to be fed into LLMs.

4.3 Automatic Metric Evaluation

4.3.1 Evaluation Setup

For automatic evaluation metrics, we adopt (1) **BERTScore**[7] (Zhang et al., 2019): This metric leverages BERT-based embedding to measure semantic similarity between predicted and reference texts. (2) **ROUGE-L** (Lin, 2004): This metric evaluates the longest common subsequence between the generated and reference texts, providing a measure of similarity in terms of sequential matching. For both metrics, higher scores indicate a better match between the predicted and the reference text.

4.3.2 Result Analysis

We conduct comprehensive experiments on the four academic writing tasks, and the results w.r.t. BERTScore and RougeL are presented in Table 3 and 4, respectively. Note that we do not conduct experiments on -M settings because its context length is too long for most of our selected baselines.

Diverse Task Difficulties and Abstractions. The four tasks we proposed are designed to challenge LLMs over long-context generation tasks with different abstraction levels. From Table 3 and 4, we can clearly observe that it provides different difficulties for LLMs to perform well from Title Writing to Related Work Writing tasks, and the results of all baselines on these four tasks have a relatively obvious trend.

[5] https://www.together.ai/

[6] https://www.langchain.com/

[7] We use deberta-xlarge-mnli (He et al., 2021) instead of the default roberta-large (Liu et al., 2019) as the backbone model to have the best correlation with human evaluation.

Table 4: **Main Results on AcademicEval w.r.t. RougeL.**

Models	**Standard LLMs**		**Long-context LLMs**			**RALM**	
	Gemma	LLaMA	Qwen	Mixtral	Hermes	Gemma†	LLaMA†
#Params.	7B	70B	72B	8x7B	8x7B	7B	70B
Context Length	8K	8K	32K	32K	32K	8K	8K
Setting: Title Writing							
Title-10K	44.5	47.1	44.2	45.2	46.2	42.7	**47.3**
Title-30K	-	-	44.5	44.6	45.9	42.6	**47.3**
Title-31K-G	-	-	44.2	44.4	45.3	42.5	**47.0**
Setting: Abstract Writing							
Abs-9K	22.4	25.0	24.3	24.1	**26.1**	23.4	24.2
Abs-28K	-	-	23.3	24.7	**26.6**	23.1	24.1
Abs-29K-G	-	-	23.3	24.9	**26.6**	23.2	24.0
Setting: Introduction Writing							
Intro-8K	14.9	**18.1**	16.2	17.2	17.8	15.4	17.9
Intro-28K	-	-	16.3	17.5	17.5	15.3	**17.8**
Intro-28K-G	-	-	16.3	17.5	17.5	15.4	**17.8**
Setting: Related Work Writing							
Related-34K	13.5	14.9	**16.0**	13.4	15.1	14.1	15.3
Related-53K	-	-	-	-	-	14.0	**15.3**
Related-53K-G	-	-	-	-	-	14.0	**15.2**

Bold indicates the highest score in each row.

† denotes augmentation with a retriever (Default: Contriever).

"-" means that the context length is too long to be fed into LLMs.

For example, the Title Writing task tends to have a higher score than the Abstract Writing task, which may indicate that the Title Writing task is easier than the Abstract Writing task. Since a title only has a few words, LLMs only need to generate a roughly related theme to achieve a high semantic similarity, while an abstract requires a more detailed description to achieve it.

Baseline Performance Comparison. Across automatic metrics, RALM with LLaMA frequently attains the highest scores in multiple settings (*e.g.*, Title-30K/31K-G, Intro-28K/28K-G, and Related-53K/53K-G), despite using an 8K input window. Standard LLMs remain competitive and long-context LLMs (*e.g.*, Qwen, Hermes) lead in some settings (*e.g.*, Related-34K on BERTScore). This exposes the shortcomings of long-context LLMs' generation capabilities, which are well revealed by AcademicEval. Among long-context LLMs, Hermes performs best overall, but is still slightly inferior to RALM with LLaMA. This shows that although the current long-context LLMs have a longer context window size, they still have great deficiencies in processing long text information. Overall, RALM often has an edge under automatic metrics, likely because retrieval concentrates salient content into shorter chunks, thereby maximizing overlap-oriented scores.

Impact of Context Length. The impact of context length on performance is evident across all task settings and both metrics, with baselines *often* performing worse as the context length increases, though the extent is model- and task-dependent. For example, the Title Writing task shows a noticeable drop in scores as the context length extends from 10K to 31K tokens. This trend is also apparent in Abstract Writing and Introduction Writing, where longer contexts correlate with decreased model performance, showing that our benchmark challenges LLMs in effectively processing ultra-long inputs.

Table 5: **Additional results on AcademicEval w.r.t. LLM-as-a-Judge win rate (%).**

Models	**Standard LLMs**		**Long-context LLMs**			**RALM**	
	Gemma	LLaMA	Qwen	Mixtral	Hermes	Gemma†	LLaMA†
#Params.	7B	70B	72B	8x7B	8x7B	7B	70B
Context Length	8K	8K	32K	32K	32K	8K	8K
Setting: Title Writing							
TITLE-10K	45.7	42.7	63.2	43.1	**72.0**	50.0	43.9
TITLE-30K	-	-	54.5	45.6	**62.5**	47.5	44.9
TITLE-31K-G	-	-	52.4	**62.7**	45.4	47.7	43.7
Setting: Abstract Writing							
ABS-9K	12.0	55.5	**77.0**	70.0	61.1	14.3	43.2
ABS-28K	-	-	**72.7**	66.1	41.2	12.7	42.0
ABS-29K-G	-	-	**71.0**	65.9	40.7	12.0	43.9
Setting: Introduction Writing							
INTRO-8K	34.6	63.2	**79.3**	61.5	58.0	48.8	64.1
INTRO-28K	-	-	**70.3**	60.1	56.9	46.5	62.9
INTRO-28K-G	-	-	**70.9**	61.9	59.3	48.2	63.9
Setting: Related Work Writing							
RELATED-34K	55.9	**91.9**	91.2	65.6	88.6	71.3	89.8
RELATED-53K	-	-	-	-	-	72.5	**90.7**
RELATED-53K-G	-	-	-	-	-	71.7	**90.2**

Bold indicates the highest score in each row.

† denotes augmentation with a retriever (Default: Contriever).

"-" means that the context length is too long to be fed into LLMs.

Impact of Few-shot Demonstrations. From Table 3 and 4, we can observe that the integration of few-shot demonstrations yields mixed effects: in several settings it is neutral or slightly negative under automatic metrics, yet correlated demonstrations can produce small but consistent gains for certain model–task pairs. This shows that current LLMs cannot exploit long few-shot demonstrations to benefit the target tasks well, emphasizing the importance of evaluating long in-context learning in LLM benchmarks. In addition, we can also find that few-shot demonstrations from co-author papers generally have a more positive impact on task performance than randomly selected ones.

4.4 LLM-as-a-Judge Evaluation

4.4.1 Evaluation Setup

To complement automatic metrics, we further incorporate an **LLM-as-a-Judge** evaluation to capture higher-level qualitative aspects beyond semantic overlap. Specifically, we employ the open-source Mixtral-8x22B-Instruct-v0.1 (Jiang et al., 2024) to assess five dimensions of generation quality: (1) *Novelty* — the degree to which the content introduces new and meaningful ideas; (2) *Feasibility* — the plausibility and practicality of the described methods or claims; (3) *Consistency* — the internal logical coherence of the output; (4) *Factuality* — the correctness of factual statements; and (5) *Academic Style* — the alignment with conventions of scholarly writing, enabling a more nuanced evaluation of LLM outputs. For each task, we report the *win rate* (%), *i.e.*, the percentage of cases where the generated text is preferred over the reference according to the LLM judge. The detailed LLM-as-a-Judge prompt can be found in Appendix D.

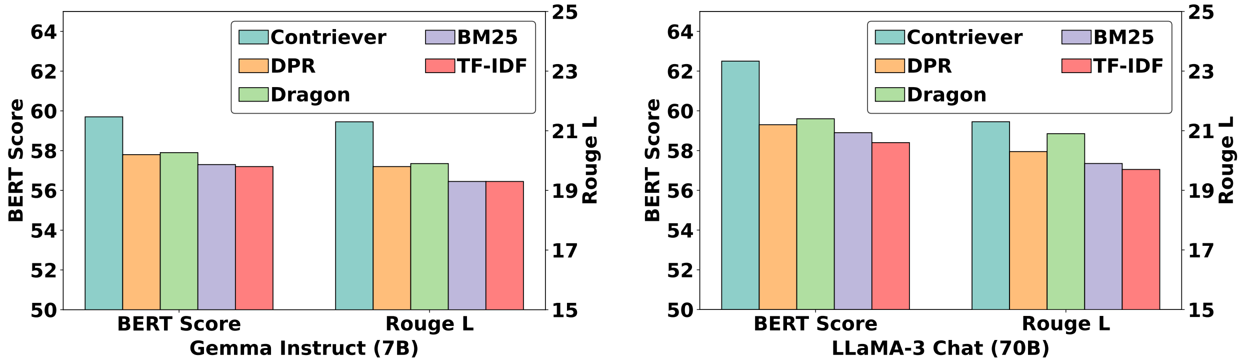

Figure 3: **Analysis of RALM on Abs-9K.** The left figure shows results with Gemma Instruct (7B), while the right one shows results with LLaMA-3 Chat (70B).

4.4.2 Result Analysis

We report results for the overall preference in Table 5. Compared to BERTScore and ROUGE-L, the LLM-as-a-Judge evaluation reveals partially different patterns, reflecting that it targets broader qualitative aspects beyond lexical or semantic overlap.

Title Writing. Under Title-10K, Hermes achieves the highest win rate (72.0), while Mixtral becomes the top model under the correlated setting Title-31K-G (62.7). This suggests that (1) short, highly abstract outputs benefit from strong style and concision (favoring Hermes at 10K); and (2) correlated contexts can help certain models (*e.g.*, Mixtral) at longer lengths when the judge considers qualities beyond surface similarity. Notably, RALM variants (Gemma†, LLaMA†) are not consistently preferred in the title task, indicating that aggressive retrieval does not always align with judged title quality.

Abstract Writing. Qwen attains the highest win rate at Abs-9K (77.0) and remains strong at longer lengths (Abs-28K: 72.7; Abs-29K-G: 71.0). Mixtral follows closely, while RALM variants trail in preference. This contrasts with automatic metrics where RALM often ranks highly, suggesting that, for abstracts, the judge values holistic qualities (coherence, feasibility, academic style) over high overlap with references.

Introduction Writing. Qwen consistently leads (Intro-8K: 79.3; Intro-28K: 70.3; Intro-28K-G: 70.9), and Hermes improves slightly with correlated contexts (56.9 → 59.3). LLaMA† remains competitive but is not top-ranked. Overall, correlated few-shot demonstrations offer modest gains for some models, supporting that graph-informed contexts can help introductions when evaluated on broader quality dimensions.

Related Work Writing. In contrast to the above tasks, RALM models (especially LLaMA†) achieve top preferences at longer lengths (Related-53K: 90.7; Related-53K-G: 90.2), aligning with the intuition that retrieval is particularly beneficial for Related Work, where judged quality rewards appropriate citations, prior studies, and domain-specific terminology.

Takeaways. (1) The judge-based preferences are *not* dominated by RALM across all tasks; instead, preferences depend on task nature and qualitative dimensions. (2) Correlated contexts can yield improvements in several settings (e.g., Mixtral on Title-31K-G, Hermes on Intro-28K-G), though gains are model-dependent. (3) The divergence from automatic metrics underscores their complementarity: automatic metrics reward overlap, whereas the judge emphasizes higher-level writing quality.

4.5 Discussion

Additional Analysis on RALM. We conduct extensive experiments on RALM on the Abs-9K setting using standard LLMs Gemma Instruct (7B) and LLaMA-3 Chat (70B), and the results are presented in Figure 3. We can find that the performance of dense retrievers consistently outperforms sparse retrievers,

Table 6: **Title-only ablation on Abstract Writing (Abs-9K).** The clear degradation indicates genuine reliance on external context rather than in-weights memorization.

Setting	**Model**	**BERTScore**	**ROUGE-L**
Default (Abs-9K)	LLaMA	62.4	25.0
Title-only (Abs-9K)	LLaMA	57.4	18.8
Default (Abs-9K)	Hermes	62.2	26.1
Title-only (Abs-9K)	Hermes	56.7	19.3

among which contriever achieves the best results. This is because the summary generation task emphasizes semantic similarity, which can be well measured by the similarity of dense embeddings. However, the sparse retrievers perform text chunk recall based on sparse embeddings, and the results are significantly worse than those of the dense retrievers.

Understanding the Performance Plateau of AcademicEval. The performance plateau observed at longer contexts (e.g., 9K→30K) invites further examination of its underlying causes. While our analysis attributes this plateau partly to the limited ability of current models to *utilize* ultra-long inputs through **in-context learning (ICL)**, another plausible factor lies in **in-weights learning (IWL)** (Chan et al., 2024). That is, certain academic knowledge may already be internalized during pretraining. In such cases, adding more context brings diminishing informational returns even when the benchmark itself remains well-constructed.

To better understand this phenomenon, we analyze both structural and empirical evidence. Structurally, AcademicEval organizes tasks across hierarchical abstraction levels (Title→Abstract→Introduction→Related Work), where deeper contextual reasoning becomes increasingly essential. Plateaus may thus occur when longer inputs introduce redundancy rather than new cues, suggesting ICL saturation instead of pure memorization. Empirically, we conduct a *Title-only* ablation on the Abstract Writing task under Abs-9K, where most contextual information is removed except for the paper title. As shown in Table 6, the BERTScore and ROUGE-L of both LLaMA and Hermes drop sharply (by 5–7 points), confirming that model performance depends strongly on the provided context and is not solved by IWL alone.

Overall, our evidence indicates that the observed plateau on AcademicEval is *primarily driven by imperfect long-context utilization* (**ICL** limitation), rather than by **IWL**. This reading is supported by the *Title-only* ablation, where removing most contextual information yields substantial drops in both BERTScore and ROUGE-L, indicating strong dependence on the provided context. Moreover, since AcademicEval is designed as a *live-updating benchmark* that continuously incorporates newly published arXiv papers via the co-author graph, the evaluation set evolves over time and reduces the likelihood that performance is dominated by pre-encoded (in-weights) knowledge. While diminishing informational returns can occur when additional tokens introduce redundancy, AcademicEval serves as a diagnostic lens showing that the plateau chiefly reflects current models' limited ability to exploit ultra-long inputs under realistic, evolving conditions.

This discussion also aims to inspire further reflection in the long-context benchmarking community on how dataset design and periodic updates can better disentangle ICL and IWL effects, while underscoring the continued importance of mitigating data leakage in future benchmark construction.

5 Conclusion

In this paper, we propose AcademicEval, a live long-context LLM benchmark for evaluating long-context generation tasks with hierarchical abstraction levels. AcademicEval adopts arXiv as the data source and introduces several long-context academic writing tasks without manual annotation since the papers on arXiv can be regarded as original, high-quality, and expert-curated labels. Moreover, we integrate few-shot demonstrations from a collected co-author graph to make the context length of our benchmark flexible and scalable. An efficient live evaluation is also designed to make AcademicEval immune to the label leakage