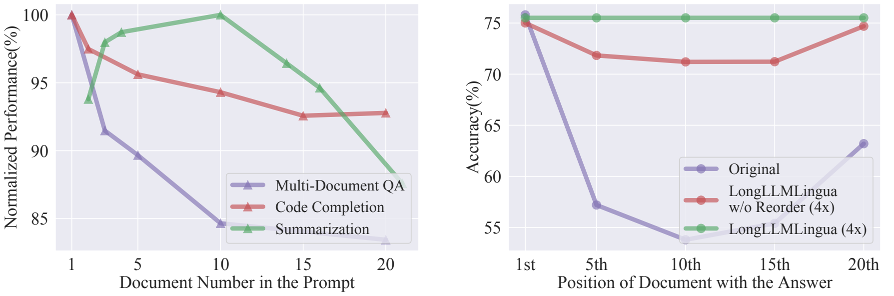
LongLLMLingua: ACCELERATING AND ENHANCING LLMS IN LONG CONTEXT SCENARIOS VIA PROMPT COMPRESSION

Huiqiang Jiang, Qianhui Wu, Xufang Luo,
Dongsheng Li, Chin-Yew Lin, Yuqing Yang, Lili Qiu
Microsoft Corporation
`{hjiang,qianhuiwu,xufang.luo}@microsoft.com`

ABSTRACT

In long context scenarios, large language models (LLMs) face three main challenges: higher computational cost, performance reduction, and position bias. Research indicates that LLM performance hinges on the density and position of key information in the input prompt. Addressing this, we introduce LongLLMLingua, a method for prompt compression that improves LLMs' key information perception, effectively tackling these challenges. Our extensive evaluation across various long context scenarios demonstrates that LongLLMLingua not only enhances performance but also significantly reduces costs and latency. For instance, in the NaturalQuestions benchmark, LongLLMLingua boosts performance by up to 21.4% with around 4x fewer tokens in GPT-3.5-Turbo, leading to substantial cost savings. It achieves a 94.0% cost reduction in the LooGLE benchmark. Moreover, when compressing prompts of about 10k tokens at ratios of 2x-6x, LongLLMLingua can accelerate end-to-end latency by 1.4x-2.6x.

1 INTRODUCTION



(a) Performance v.s. Document Number (b) Performance v.s. Key Information Position

Figure 1: (a) LLMs' performance in downstream tasks tends to decline with the increase of noisy information in the prompt. In this case, we keep k most relevant documents/paragraphs based on the ground-truth or LongLLMLingua r_k. A larger k implies more noise introduced into the prompt. To improve the key information density in the prompt, we present question-aware coarse-to-fine compression. (b) LLMs' ability to capture the relevant information depends on their positions in the prompt. To reduce information loss in the middle, we introduce a document reordering mechanism.

ChatGPT and other large language models (LLMs) are pivotal in advancing user-oriented language technologies across various applications. Optimal prompt design is key for enhanced performance in specific tasks. Technologies like In-Context Learning (ICL) (Dong et al., 2023), Retrieval Augmented Generation (RAG) (Lewis et al., 2020), and Agent (Park et al., 2023) lead to increasingly lengthy prompts, sometimes extending to thousands of tokens. Scenarios such as multi-document

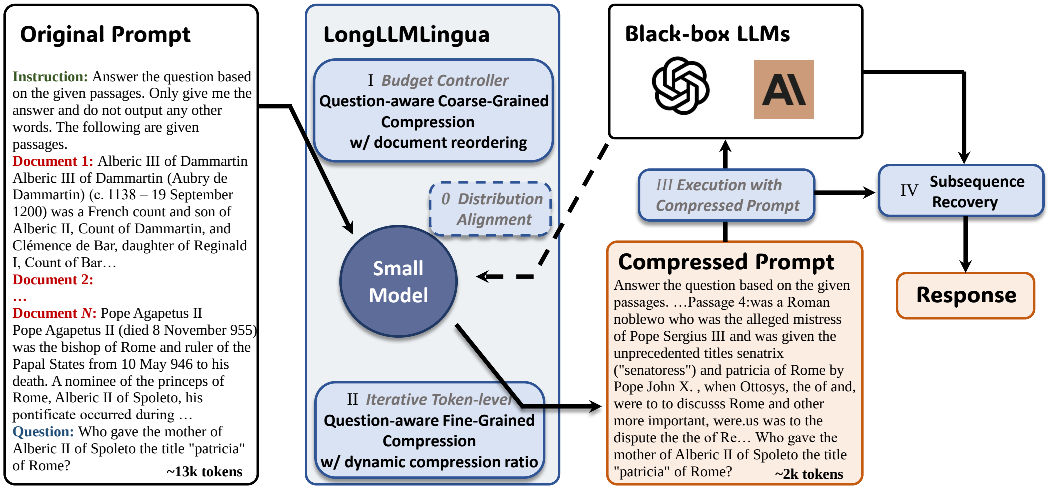


Figure 2: Framework of *LongLLMLingua*. Gray *Italic* content: As in LLMLingua.

question answering, code completion, and document summarization also necessitate the processing of long contexts.

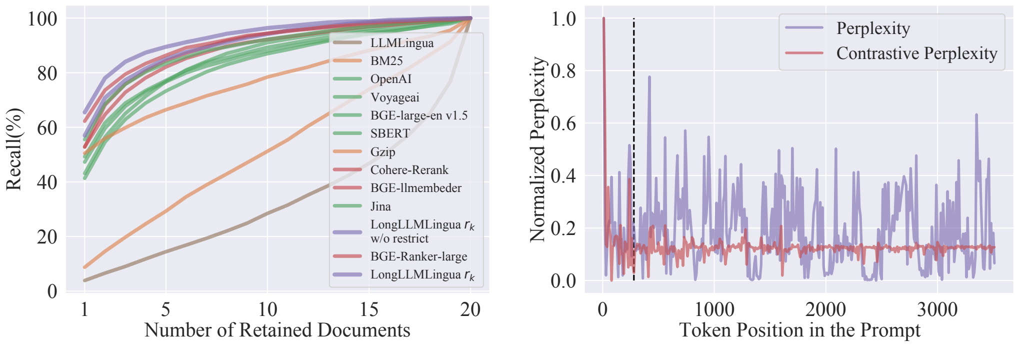
There are three main challenges when LLMs are used in long context scenarios: (1) Higher computational costs, encompassing both financial and latency expenses. (2) Longer prompts introduce irrelevant and redundant information, which can weaken LLMs' performance (Shi et al., 2023), as illustrated in Figure 1a. (3) LLMs exhibit position bias, also known as the 'lost in the middle' issue (Liu et al., 2024; Kamradt, 2023), suggesting that the placement of key information within the prompt significantly affects LLMs' performance. This is demonstrated by the purple curve in Figure 1b.

Inspired by these observations, we propose *LongLLMLingua* to address the three challenges. Specifically, we use LLMLingua (Jiang et al., 2023a) as our backbone framework for prompt compression to address the first challenge, *i.e.*, reduce cost and latency. However, in the case of long contexts, the distribution of question-relevant key information in the prompt is generally dynamic and sparse. Existing prompt compression methods like LLMLingua (Jiang et al., 2023a) and Selective-Context (Li et al., 2023c) that often fail to consider question during compression, resulting in retention of excessive noise and decreased performance. LongLLMLingua aims to improve LLMs' perception of key information pertinent to the question, thereby overcoming the noise and position bias issues in long contexts, shown in Figure 1b. The underlying principle of LongLLMLingua is that small language models are inherently capable of capturing the distribution of key information relevant to a given question.

Our main contributions include: (1) We propose a question-aware coarse-to-fine compression method to improve the key information density in the prompt (Sec. 2.1); (2) We introduce a document reordering strategy to minimize position bias in LLMs. (Sec. 2.2); (3) We establish dynamic compression ratios for precise control between coarse and fine compression levels (Sec. 2.3); (4) We propose a post-compression subsequence recovery strategy to improve the integrity of the key information (2.4). (5) We evaluate LongLLMLingua across five benchmarks, *i.e.*, NaturalQuestions (Liu et al., 2024), LongBench (Bai et al., 2023), ZeroSCROLLS (Shaham et al., 2023), MuSicQue (Trivedi et al., 2022), and LooGLE (Li et al., 2023b), covering a variety of long context scenarios. Experimental results reveal that LongLLMLingua's compressed prompts outperform original prompts in terms of performance, cost efficiency, and system latency.

2 LONGLLMLINGUA

LongLLMLingua is developed upon the framework of LLMLingua (Jiang et al., 2023a) towards prompt compression in long context scenarios. To address the three challenges in long context scenarios mentioned in Sec. 1, we propose LongLLMLingua. This method aims to enhance the



(a) Recall Distribution

(b) Perplexity Distribution

Figure 3: (a) Comparison of recall on NaturalQuestions Multi-documemnt QA dataset, which increases from top to bottom in terms of Recall@1. Different colors represent different types of methods. Among them, yellow represents traditional relevance methods, green signifies embedding-based methods, and red denotes rerank-based methods. (b) Comparison between perplexities and contrastive perplexities of tokens in the prompt from Multi-documemnt QA dataset. The document with the ground truth is located on the left side of the dashed line.

perception of LLMs of key information relevant to the question within the prompt. It encompasses three perspectives and further incorporates a subsequence recovery strategy to enhance the accuracy and reliability of the information provided to users. We elaborate on each component in this section.

2.1 HOW TO IMPROVE KEY INFORMATION DENSITY IN THE PROMPT?

Question-Aware Coarse-Grained Compression In coarse-grained compression, we aim to figure out a metric r_k to evaluate the importance of each document $\mathbf{x}_k^{\text{doc}} = \{x_{k,i}^{\text{doc}}\}_{i=1}^{N_k}$, where N_k is the number of tokens in $\mathbf{x}_k^{\text{doc}}$. We only keep $\mathbf{x}_k^{\text{doc}}$ with higher r_k as the intermediate compressed results.

We propose to use the perplexity of the question $\mathbf{x}^{\text{que}}$ conditioned on different contexts $\mathbf{x}_k^{\text{doc}}$ to represent the association between them. We also append a restrictive statement $\mathbf{x}^{\text{restrict}}$ [1] after $\mathbf{x}^{\text{que}}$ to strengthen the interconnection of $\mathbf{x}^{\text{que}}$ and $\mathbf{x}_k^{\text{doc}}$. It can be regarded as a regularization term that mitigates the impact of hallucinations. This can be formulated as:

$$r_k = -\frac{1}{N_c}\sum_{i}^{N_c} \log p(x_i^{\text{que,restrict}}|\mathbf{x}_k^{\text{doc}}), k \in \{1, 2, \cdots, K\}, \tag{1}$$

where $x_i^{\text{que,restrict}}$ is the i-th token in the sequence of $\mathbf{x}^{\text{que}}$ and $\mathbf{x}^{\text{restrict}}$ and N_c in the number of tokens.

Figure 3a displays the recall distribution of different retrieval methods, including traditional relevance methos (BM25, Gzip (Jiang et al., 2023b)), embedding-based methods(OpenAI-embedding, Voyageai[2], BGE-large-en v1.5 (Xiao et al., 2023), Sentence-BERT (Reimers & Gurevych, 2019), Jina (Günther et al., 2023)), and reranker methods (Cohere-Rerank[3], BGE-llmembeder, BGE-Ranker-large). It can be observed that our methods achieve the best recall across various numbers of retrieved documents.

LLMLingua uses document-level perplexity to represent the importance of documents:$r_k = 1/N_k \sum_i^{N_k} p(x_{k,i}^{\text{doc}}) \log p(x_{k,i}^{\text{doc}}), k \in \{1, 2, \cdots, K\}$. Although the retained documents typically contain a lot of information, they are irrelevant to the question $\mathbf{x}^{\text{que}}$ and instead become noise, reducing key information density in the compressed results and bringing difficulties for LLM to output cor-

---

[1] Specifically, "*We can get the answer to this question in the given documents*".

[2] https://www.voyageai.com/

[3] https://cohere.com/rerank

rect answers. As shown in Figure 3a, the recall@16 of LLMLingua only reaches 50%, indicating its incompetence in retaining key information during compression.

Retrieval-based methods are also feasible here. We can use $\mathbf{x}^{\text{que}}$ to retrieve the most relevant documents among $(\mathbf{x}_1^{\text{doc}}, \cdots, \mathbf{x}_K^{\text{doc}})$ as the compressed results. However, these methods struggle to distinguish question-related fine-grained semantic information. Some documents with key information may be discarded during retrieval. As shown in Figure 3a, embedding-based methods such as Sentence BERT and OpenAI Embedding only achieve ∼75% accuracy in recall@5, which implies that the final accuracy upper bound of LLMs with 4x compression is only 75%.

Question-Aware Fine-Grained Compression In this paper, we propose *contrastive perplexity*, *i.e.*, the distribution shift caused by the condition of the question, to represent the association between the token and the question. The contrastive perplexity based importance metric s_i for each token x_i in $\{\mathbf{x}_k^{\text{doc}}\}_{k=1}^{K'}$ can be formulated follow. Additionally, we provide the derivation of its mathematical significance in the Appendix B, concluding that it is equivalent to conditional pointwise mutual information (Church & Hanks, 1989).

$$s_i = \text{perplexity}(x_i|x_{<i}) - \text{perplexity}(x_i|x^{\text{que}}, x_{<i}). \tag{2}$$

Figure 3b illustrates the difference between perplexities and contrastive perplexities. We can see that tokens of high perplexities are widely distributed in all documents. However, tokens with high contrastive perplexities concentrate more on the left side of the dashed line, which corresponds to the document that contains the answer to the question. This suggests that the proposed contrastive perplexity can better distinguish tokens relevant to the question, thus improving the key information density in the compressed results. Moreover, in the Appendix C, we present the sentence-level empirical evidence of question-aware token-level prompt compression method, demonstrating its effectiveness.

2.2 HOW TO REDUCE INFORMATION LOSS IN THE MIDDLE?

After the coarse-grained compression, we have obtained a set of documents $\{\mathbf{x}_k^{\text{doc}}\}_{k=1}^{K'}$ with their corresponding importance scores $\{r_k\}_{k=1}^{K'}$ indicating their association with the question $\mathbf{x}^{\text{que}}$. Therefore, we reorder documents using their importance scores to better leverage LLMs' information perception difference in positions as demonstrated in Figure 1b:

$$(\mathbf{x}^{\text{ins}}, \mathbf{x}_1^{\text{doc}}, \cdots, \mathbf{x}_{K'}^{\text{doc}}, \mathbf{x}^{\text{que}}) \xrightarrow{r_k} (\mathbf{x}^{\text{ins}}, \mathbf{x}_{r1}^{\text{doc}}, \cdots, \mathbf{x}_{rK'}^{\text{doc}}, \mathbf{x}^{\text{que}}) \tag{3}$$

2.3 HOW TO ACHIEVE ADAPTIVE GRANULAR CONTROL DURING COMPRESSION?

We bridge coarse-grained compression to fine-grained compression and use the importance scores $\{r_k\}_{k=1}^{K'}$ obtained from coarse-grained compression to guide the budget allocation in fine-grained compression. In this way, we can achieve adaptive granular control on the whole. Specifically, we first determine the initial budget for the retained documents τ^{doc} [4] using the budget controller of LLMLingua. During fine-grained compression, we follow the iterative token-level compression algorithm in LLMLingua but dynamically assign the compression budget τ_k^{doc} to each document $\mathbf{x}_k^{\text{doc}}$ according to the ranking index $I(r_k)$ (e.g., 0, 1) of its importance score from the coarse-grained compression. In this paper, we employ a linear scheduler for the adaptive allocation. Budget of each token x_i can be formulated as:

$$\begin{aligned} \tau_i &= \tau_k^{\text{doc}}, \quad \forall x_i \in \mathbf{x}_k^{\text{doc}}, \\ \tau_k^{\text{doc}} &= \max(\min((1 - \frac{2I(r_k)}{K'})\delta\tau + \tau^{\text{doc}}, 1), 0), \end{aligned} \tag{4}$$

where i and k is the index of token and document, K' denotes the number of documents, and $\delta\tau$ is a hyper-parameter that controls the overall budget for dynamic allocation.

---

[4] In LLMLingua, it is τ^{dems} for demonstrations.

Table 1: Performance of different methods with 4x compression ratio (raw size / compressed size = $1/\tau$) on NaturalQuestions (20 docs.) (Liu et al., 2024). Reorder: we reorder the documents with relevance metrics of different baselines as our document reordering strategy described in Sec. 2.2.

Methods	GPT3.5-Turbo						LongChat-13b						Length		Latency	
	1st	5th	10th	15th	20th	Reorder	1st	5th	10th	15th	20th	Reorder	Tokens	$1/\tau$	Latency	Speedup
Retrieval-based Methods																
BM25	40.6	38.6	38.2	37.4	36.6	36.3	39.5	37.5	36.8	36.4	35.5	37.7	798	3.7x	1.5	2.7x
Gzip	63.1	61.0	59.8	61.1	60.1	62.3	57.6	52.9	51.0	50.1	50.4	57.2	824	3.6x	1.5	2.7x
SBERT	66.9	61.1	59.0	61.2	60.3	64.4	62.6	56.6	55.1	53.9	55.0	59.1	808	3.6x	1.6	2.5x
OpenAI	63.8	64.6	65.4	64.1	63.7	63.7	61.2	56.0	55.1	54.4	55.0	58.8	804	3.7x	4.3	1.0x
LongLLMLingua r_k	71.1	70.7	69.3	68.7	68.5	71.5	67.8	59.4	57.7	57.7	58.6	64.0	807	3.7x	1.7	2.4x
Compression-based Methods																
Selective-Context	31.4	19.5	24.7	24.1	43.8	-	38.2	17.2	15.9	16.0	27.3	-	791	3.7x	6.8	0.6x
LLMLingua	25.5	27.5	23.5	26.5	30.0	27.0	32.1	30.8	29.9	28.9	32.4	30.5	775	3.8x	1.8	2.2x
LongLLMLingua	**75.0**	**71.8**	**71.2**	**71.2**	**74.7**	**75.5**	**68.7**	**60.5**	**59.3**	**58.3**	**61.3**	**66.7**	748	3.9x	2.1	2.0x
Original Prompt	75.7	57.3	54.1	55.4	63.1	-	68.6	57.4	55.3	52.5	55.0	-	2,946	-	4.1	-
Zero-shot	56.1						35.0						15	196x	1.1	3.7x

2.4 HOW TO IMPROVE THE INTEGRITY OF KEY INFORMATION?

We propose a subsequence recovery method to restore the original content from LLMs' responses. This method relies on the subsequence relationship among tokens in the original prompt, compressed prompt, and LLMs' response. The overall procedure includes: i) Iterate through tokens y_l in LLMs' response and select the longest substring $\widetilde{\boldsymbol{y}}_{\text{key},l} = \{y_l, y_{l+1}, ..., y_r\}$ that appears in the compressed prompt $\widetilde{\boldsymbol{x}}$. ii) Find the maximum common shortest subsequence $\boldsymbol{x}_{i,j} = \{x_i, x_{i+1}, ..., x_j\}$ in the original prompt $\boldsymbol{x}$, corresponding to the representation $\widetilde{\boldsymbol{y}}_{\text{key},l}$ in the original prompt (accelerated using prefix trees or sequence automata). iii) Replace the matched tokens $\widetilde{\boldsymbol{y}}_{\text{key},l}$ in LLMs' response with the corresponding subsequence $\boldsymbol{x}_{i,j}$ from the original prompt. Appendix A show more details.

3 EXPERIMENTS

Table 1 and Appendix E showcase the performance and latency of various methods under different compression constraints across multiple tasks, including multi-document QA (Liu et al., 2024), multi-hop QA (Trivedi et al., 2022), and long context benchmarks (LongBench (Bai et al., 2023), ZeroScROLLS (Shaham et al., 2023), LooGLE (Li et al., 2023b)). Table 5 and 2 presents the ablation results of our proposed module in LongLLMLingua. Additionally, we display some cases to demonstrate the results of prompt compression by the modules designed in LongLLMLingua across different scenarios in Appendix H and I.

There are multiple observations and conclusions: (1) LongLLMLingua consistently outperforms across various tasks and compression ratios, enhancing performance while significantly reducing costs. For instance, it boosts performance by 21.4% on NaturalQuestions using only a quarter of the tokens. (2) Compression methods like Selective Context (Li et al., 2023c) and LLMLingua (Jiang et al., 2023a) often underperform, particularly in tasks with excessive irrelevant information. Their information entropy-based compression includes too much noise, sometimes leading to worse outcomes than zero-shot settings. (3) Retrieval-based methods excel at low compression ratios but decline with higher compression, due to reduced recall shown in Figure 3a. (4) LongLLMLingua and our coarse-grained compression metric r_k outperform all other baselines across various tasks and compression ratios. Notably, as compression ratios rise, say from 2x to 4x, LongLLMLingua even shows a slight increase in performance. This success is due to our question-aware coarse-to-fine method that effectively pinpoints key information. (5) The proposed reordering method helps in not only our approach but also other baselines as shown in Table 1, well demonstrating its effectiveness.

4 CONCLUSION

We propose LongLLMLingua to address the challenges of long contexts through question-aware prompt compression and document reordering. The experimental results demonstrate the effectiveness and efficiency of our method, which can enhance the information density within the prompt, mitigate the 'lost in the middle' issue, and boost the performance of LLMs.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv preprint*, abs/2308.14508, 2023. URL `https://arxiv.org/abs/2308.14508`.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R. Gormley. Unlimiformer: Long-range transformers with unlimited length input. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=lJWUJWLCJo`.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=JroZRaRw7Eu`.

Harrison Chase. LangChain, 2022. URL `https://github.com/hwchase17/langchain`.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *ArXiv preprint*, abs/2306.15595, 2023. URL `https://arxiv.org/abs/2306.15595`.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.232. URL `https://aclanthology.org/2023.emnlp-main.232`.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *27th Annual Meeting of the Association for Computational Linguistics*, pp. 76–83, Vancouver, British Columbia, Canada, 1989. Association for Computational Linguistics. doi: 10.3115/981623.981633. URL `https://aclanthology.org/P89-1010`.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens. *ArXiv preprint*, abs/2307.02486, 2023. URL `https://arxiv.org/abs/2307.02486`.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *ArXiv preprint*, abs/2301.00234, 2023. URL `https://arxiv.org/abs/2301.00234`.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=uREj4ZuGJE`.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan T. Chakaravarthy, Yogish Sabharwal, and Ashish Verma. Power-bert: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3690–3699. PMLR, 2020. URL `http://proceedings.mlr.press/v119/goyal20a.html`.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *ArXiv preprint*, abs/2310.19923, 2023. URL `https://arxiv.org/abs/2310.19923`.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Simple on-the-fly length generalization for large language models. *ArXiv preprint*, abs/2308.16137, 2023. URL `https://arxiv.org/abs/2308.16137`.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=jKN1pXi7b0`.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.emnlp-main.825. URL `https://aclanthology.org/2023.emnlp-main.825`.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6810–6828, Toronto, Canada, 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.426. URL `https://aclanthology.org/2023.findings-acl.426`.

Greg Kamradt. Needle In A Haystack - Pressure Testing LLMs, 2023. URL `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

Gyuwan Kim and Kyunghyun Cho. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6501–6511, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.508. URL `https://aclanthology.org/2021.acl-long.508`.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL `https://aclanthology.org/Q19-1026`.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html`.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can open-source llms truly promise on context length?, 2023a. URL `https://lmsys.org/blog/2023-06-29-longchat`.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. Loogle: Can long-context language models understand long contexts? *ArXiv preprint*, abs/2311.04939, 2023b. URL `https://arxiv.org/abs/2311.04939`.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353, Singapore, 2023c. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.391. URL `https://aclanthology.org/2023.emnlp-main.391`.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 02 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00638. URL `https://doi.org/10.1162/tacl_a_00638`.

Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. AdapLeR: Speeding up inference by adaptive length reduction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1–15, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.1. URL `https://aclanthology.org/2022.acl-long.1`.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=2DtxPCL3T5`.

Erik Nijkamp, Tian Xie, Hiroaki Hayashi, Bo Pang, Congying Xia, Chen Xing, Jesse Vig, Semih Yavuz, Philippe Laban, Ben Krause, Senthil Purushwalkam, Tong Niu, Wojciech Kryściński, Lidiya Murakhovs'ka, Prafulla Kumar Choubey, Alex Fabbri, Ye Liu, Rui Meng, Lifu Tu, Meghana Bhat, Chien-Sheng Wu, Silvio Savarese, Yingbo Zhou, Shafiq Joty, and Caiming Xiong. Xgen-7b technical report. *ArXiv preprint*, abs/2309.03450, 2023. URL `https://arxiv.org/abs/2309.03450`.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL `https://doi.org/10.1145/3586183.3606763`.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=wHBfxhZu1u`.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=R8sQPpGCv0`.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410`.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7977–7989, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.536. URL `https://aclanthology.org/2023.findings-emnlp.536`.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *ArXiv preprint*, abs/2307.08621, 2023. URL `https://arxiv.org/abs/2307.08621`.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL `https://aclanthology.org/2022.tacl-1.31`.

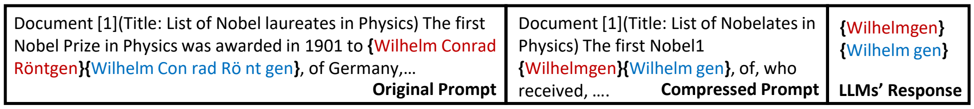
Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=s1FjXzJ0jy`.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1423–1436, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.79. URL `https://aclanthology.org/2023.acl-long.79`.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding. *ArXiv preprint*, abs/2309.07597, 2023. URL `https://arxiv.org/abs/2309.07597`.

A TOKEN-LEVEL SUBSEQUENCE RECOVERY DETAILS

Certain tokens of key entities may be discarded during the fine-grained token-wise compression. For example, the time entity "2009" in the original prompt might be compressed to "209" and the name entity "Wilhelm Conrad Röntgen" might be compressed to "Wilhelmgen". This can cause problems for fact-based tasks like document QA, where language models tend to replicate information from the prompt, as shown in Figure 4. The details of the recovery algorithm are shown in Algorithm 1.



Document [1](Title: List of Nobel laureates in Physics) The first Nobel Prize in Physics was awarded in 1901 to {Wilhelm Conrad Röntgen}{Wilhelm Con rad Rö nt gen}, of Germany,…

Original Prompt

Document [1](Title: List of Nobelates in Physics) The first Nobel1 {Wilhelmgen}{Wilhelm gen}, of, who received, ….

Compressed Prompt

{Wilhelmgen}
{Wilhelm gen}

LLMs' Response

Figure 4: The example of Subsequence Recovery, the red text represents the original text, and the blue text is the result after using the LLaMA 2-7B tokenizer.

Algorithm 1 Pseudo code of Token-level Subsequence Recovery.

```
Input: The original prompt $\boldsymbol{x}$; the compressed prompt $\widetilde{\boldsymbol{x}}$; the generation response of LLMs $\boldsymbol{y}$.
Set the final response list $\boldsymbol{y}_{\text{rec}} = \phi$, the left token index of subsequence $l$ to 0.
while $l < \boldsymbol{y}.len()$ do
    if Substring $y_l \in \widetilde{\boldsymbol{x}}$ then
        Find the longer substring $\widetilde{\boldsymbol{y}}_{\text{key},l} = \{y_l, y_{l+1}, ..., y_r\} \in \widetilde{\boldsymbol{x}}$.
        Find the maximum common shortest subsequence $\boldsymbol{x}_{i,j} = \{x_i, x_{i+1}, ..., x_j\}$ in the original prompt $\boldsymbol{x}$.
        Add the subsequence $\boldsymbol{x}_{i,j} = \{x_i, x_{i+1}, ..., x_j\}$ to the response $\boldsymbol{y}_{\text{rec}}$.
        Set the left index $l$ to $r + 1$.
    else
        Add the token $y_l$ to the response $\boldsymbol{y}_{\text{rec}}$.
        Set the left index $l$ to $l + 1$.
    end if
end while
Output: The final response list $\boldsymbol{y}_{\text{rec}}$.
```

B DERIVATION OF QUESTION-AWARE FINE-GRAINED COMPRESSION

Based on the definition of Eq. 2, we can derive that,

$$
\begin{aligned}
s_i &= \text{perplexity}(x_i|x_{<i}) - \text{perplexity}(x_i|x^{\text{que}}, x_{<i}) \\
&= q(x_i)\log p(x_i|x^{\text{que}}, x_{<i}) - q(x_i)\log p(x_i|x_{<i}) \\
&= q(x_i)\log \frac{p(x_i|x^{\text{que}}, x_{<i})}{p(x_i|x_{<i})}
\end{aligned} \tag{5}
$$

In the actual calculation of perplexity, a log operation is performed to avoid overflow, and $q(x_i)$ represents the probability distribution of the ground-truth.

At the same time, we can derive the following expanded expression based on Bayes' theorem.

$$
p(x^{\text{que}}|x_i, x_{<i}) = \frac{p(x_i|x^{\text{que}}, x_{<i})p(x^{\text{que}})}{p(x_i|x_{<i})} = p(x^{\text{que}})\frac{p(x_i|x^{\text{que}}, x_{<i})}{p(x_i|x_{<i})} \tag{6}
$$

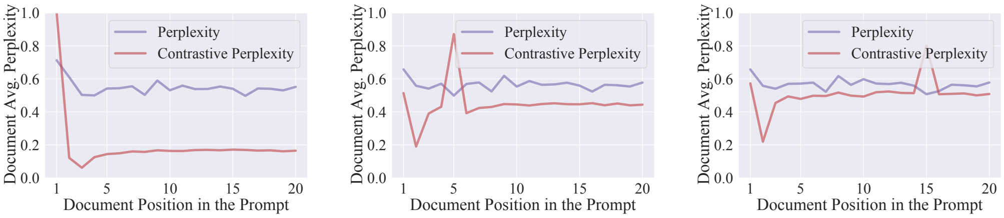
The probability distribution $p(x^{\text{que}})$ of the question and the ground-truth distribution $q(x_i)$ of x_i are constants, hence s_i can be considered as the representation of Eq. 6.

$$
s_i \propto p(x^{\text{que}}|x_i, x_{<i}) \tag{7}
$$

So we can utilize Eq. 2 to represent the probability distribution $p(x^{\text{que}}|x_i, x_{<i})$, which represents the condition likelihood of generating x^{que} given the token x_i. Therefore, we can represent the token-level sensitive distribution for the question x^{que} using just a single inference. For tokens that

are unrelated to x^{que}, such as the tokens on the right side of Figure 3b, their original amount of information may be high, but the contrastive perplexity remains at a relatively low level. Finally, we observe that the form of contrastive perplexity is equivalent to conditional pointwise mutual information (Church & Hanks, 1989).

C EMPIRICAL STUDY OF QUESTION-AWARE FINE-GRAINED COMPRESSION



(a) 1st (b) 5th (c) 15th

Figure 5: The distribution of document-level average perplexity when the ground-truth document is in different positions.

Figure 5 shows the distribution of the document's average perplexity when the ground-truth is located at different positions within the prompt. As can be observed, as the context length increases, the original perplexity curve remains relatively stable. In unrelated documents, a higher perplexity is still retained, making it easier to remove relevant tokens from the related documents in the prompt compression process, thereby damaging the corresponding semantic information. Contrarily, contrastive perplexity shows an increase in perplexity in documents related to the question. According to the theoretical derivation in Appendix B, it's known that contrastive perplexity characterizes the conditional probability of tokens corresponding to the question. The higher the relevance, the higher the contrastive perplexity, thereby retaining key information in the prompt compression process.

D EXPERIMENT DETAILS

D.1 DATASET DETAILS

We use NaturalQuestions (Liu et al., 2024) for the multi-document QA task, MuSicQue (Trivedi et al., 2022) for the multi-hop QA task, and use LongBench (Bai et al., 2023), ZeroSCROLLS (Shaham et al., 2023), LooGLE (Li et al., 2023b) for general long context scenarios. The specific details of the dataset are as follows:

NaturalQuestions multi-document QA A multi-document question-answering dataset, comprising 2,655 problems, was built by Liu et al. (2024) based on the NaturalQuestions dataset (Kwiatkowski et al., 2019). This dataset provides a realistic retrieval-augmented generation setup that closely resembles commercial search and question-answering applications (e.g., Bing Chat). Each example in the dataset contains a question and k related documents, utilizing the Contriever retrieval system (Izacard et al., 2022), one of which includes a document with the correct answer. To perform this task, the model must access the document containing the answer within its input context and use it to answer the question. The dataset's data is sourced from the NaturalQuestions dataset, which contains historical queries issued to the Google search engine and human-annotated answers extracted from Wikipedia. The average prompt token length in this benchmark is 2,946. For our experiments, we used the version provided by Liu et al. (2024) that includes 20 documents[5]. The dataset comprises five different ground truth document position settings in the prompt: 1st, 5th, 10th, 15th, and 20th.

---

[5] https://github.com/nelson-liu/lost-in-the-middle

LongBench A multi-task long context benchmark consists of 3,750 problems in English and includes six categories with a total of 16 tasks. These tasks encompass key long-text application scenarios, such as single-document QA, multi-document QA, summarization, few-shot learning, synthetic tasks, and code completion. The average prompt token length in this benchmark is 10,289. For our experiments, we used the English dataset and evaluation scripts provided by Bai et al. (2023) for this benchmark[6].

ZeroSCROLLS The multi-task long context benchmark consists of 4,378 problems, including four categories with a total of 10 tasks. These tasks cover summarization, question answering, aggregated sentiment classification, and information reordering. The average prompt token length in this benchmark is 9,788. For our experiments, we used the validation set and evaluation scripts provided by Shaham et al. (2023) for this dataset[7].

MuSiQue The multi-hop question-answer dataset is composed of 39,876, 4,834, and 4,918 problems in the training, validation, and testing datasets, respectively. This dataset requires the language model to conduct multiple inferences based on the content of several documents and provide corresponding answers, thereby necessitating a certain capability for global information processing. The average token length for prompts in this dataset is 2,477. For our experiments, we utilized the validation set and evaluation scripts provided by Trivedi et al. (2022) for this dataset[8].

LooGLE The multi-task long context benchmark comprises 6,448 problems, divided into three categories: summarization, short dependency question answering, and long dependency question answering. The average prompt token length in this benchmark stands at 24,005. For our experiments, we focused on the long dependency question answering subset, which includes four types of tasks: information retrieval, timeline reordering, computation, and comprehension. This subset contains 1,101 problems. We utilized the evaluation scripts provided by Li et al. (2023b) for this dataset[9].

D.2 BASELINES

We include two sets of baselines in following experiments:

Retrieval-based Methods We measure the association between the question and the documents in the prompt using five SoTA retrieval methods: BM25, Gzip (Jiang et al., 2023b), Sentence-BERT (Reimers & Gurevych, 2019), OpenAI Embedding, and the important metric r_k used in LongLLMLingua coarse-grained compression. We discard sentences or paragraphs with low association until the compression constraint is met while keeping the original document order unchanged.

Compression-based Methods We compare our approach with two state-of-art methods for prompt compression, *i.e.*, Selective Context (Li et al., 2023c) and LLMLingua (Jiang et al., 2023a). Both methods employ LLaMA-2-7B-Chat as the small language model for compression. In LLMLingua, a coarse-to-fine approach is used to handle constraints of compression ratio: the original prompt is first compressed to k times the constraint at a coarse level, where k is the granular control coefficient; token-level is then performed to reach the overall constraint. Our method follows the same coarse-to-fine logic to achieve the constraint.

D.3 IMPLEMENTATION DETAILS

In this paper, we use GPT-3.5-Turbo-0613[10] and LongChat-13B-16k as the target LLMs, both accessible via OpenAI[11] and HuggingFace[12]. To ensure stable and reproducible results, we employ

---

[6] https://github.com/THUDM/LongBench

[7] https://www.zero.scrolls-benchmark.com/

[8] https://github.com/stonybrooknlp/musique

[9] https://github.com/bigai-nlco/LooGLE

[10] For experiments with original prompts exceeding 4k tokens, we utilize GPT-3.5-Turbo-16k-0613.

[11] https://platform.openai.com

[12] https://huggingface.co/lmsys/longchat-13b-16k

greedy decoding and set the temperature to 0 in all experiments. For the small language models used for compression, we apply LLaMA-2-7B-Chat[13], which has been aligned by supervised fine-tuning and RLHF. We implement our approach with PyTorch 1.13.1 and HuggingFace Transformers. We set up hyperparameters following LLMLingua except for the segment size used in iterative token-level compression set to 200 here.

All experiments were conducted using a Tesla V100 (32GB). We use tiktoken[14] and GPT-3.5-Turbo model to count all the tokens. We set the granular control coefficient k to 2. We use the pre-defined compression rates $\tau_{\text{ins}} = 0.85$ and $\tau_{\text{que}} = 0.9$ for instructions and questions. The segment size used in the iterative token-level compression is set to 200. The $\delta\tau$ used in dynamic compression ratio is set to 0.25. For a fair comparison, we only used reordering in the NaturalQuestions Multi-document QA and noted this in Table 1. We use "*We can get the answer to this question in the given documents.*" as the guideline sentence in Equation (2).

For the baselines experiment, we use the currently recommended strongest model, all-mpnet-base-v2[15], as the dense representation model for SentenceBERT. We use the recommended "text-embedding-ada-002" as the embedding model for OpenAI Embedding[16]. We use the GPT2-dolly[17] as the small language model in w/ GPT2-small ablation experiments.

E Additional Experimental Results

E.1 Ablation in LongBench

Table 2: Ablation on LongBench (Bai et al., 2023) using GPT-3.5-Turbo.

Methods	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Code	AVG	Tokens	$1/\tau$
LongLLMLingua	39.0	42.2	27.4	69.3	53.8	56.6	48.0	1,809	6x
Question-aware Coarse-grained									
- w/o Question-awareness	27.1	38.7	25.4	62.0	18.0	53.3	37.4	1,945	5x
- w/ SBERT	34.0	38.7	24.1	57.9	32.5	31.1	36.4	1,790	6x
- w/ $p(\mathbf{x}_k^{\text{doc}}\|x_i^{\text{que,restrict}})$	22.5	28.9	23.2	53.0	22.5	33.3	30.6	1,794	6x
- w/o restrict	37.8	39.5	26.4	64.8	52.5	55.8	46.1	1,834	6x
- w/o Question-aware Fine-grained	35.7	41.1	26.4	62.9	44.5	54.8	44.2	1,807	6x
- w/o Dynamic Compression Ratio	36.1	40.6	26.9	67.2	48.0	55.8	45.7	1,851	6x
- w/o Subsequence Recovery	38.6	41.8	27.3	69.0	53.8	56.6	47.8	1,809	6x
- w/ Document Reordering	39.9	43.2	27.4	69.8	53.0	56.7	48.3	1,822	6x
- w/ GPT2-small	35.9	39.4	25.0	60.6	42.0	55.4	43.0	1,892	5x

Table 2 presents the results from the ablation experiment in the LongBench long context benchmark. It can be observed that in various long context tasks: 1) Removing the question-aware coarse-grained, question-aware fine-grained, dynamic compression ratio, document reordering, and subsequence recovery proposed by LongLLMLingua all result in different degrees of performance drop. 2) Among these, question-aware coarse-grained is particularly important for document-based QA and synthetic tasks, with the maximum drop being 35.8 points; its impact on summarization and code tasks is relatively smaller. 3) The design of the conditional probability in the question-aware coarse-grained module improves the results in all tasks, including code completion, single-document question-answer, and synthetic tasks. Changing the order of conditional probabilities or removing the restrict prompt both lead to varying degrees of performance decline. 4) Removing question-aware fine-grained, dynamic compression ratio has a more significant impact on document-based QA and synthetic tasks. 5) The subsequence recovery module can enhance reference-based tasks, but its improvement on tasks like summarization, code, synthetic, etc., is relatively smaller. 6) Document reordering is effective for all types of tasks. Reordering at the document level does not affect LLMs' understanding of context information, even for timeline-related tasks (see timeline reorder

---

[13] https://ai.meta.com/llama/

[14] https://github.com/openai/tiktoken

[15] https://www.sbert.net/docs/pretrained_models.html

[16] https://platform.openai.com/docs/guides/embeddings/

[17] https://huggingface.co/lgaalves/gpt2-dolly

Table 3: Performance of different methods with 2x compression ratio on NaturalQuestions (20 documents) (Liu et al., 2024). Reorder: we reorder the documents with relevance metrics of different baselines as our document reordering strategy described in Sec. 2.2. In the case of OpenAI, it corresponds to LongContextReorder in the LangChain framework (Chase, 2022). For results reported under 1st to 20th, we do not use the reordering strategy for all methods.

Methods	**GPT3.5-Turbo**						**LongChat-13b**						**Length**		**Latency**	
	1st	5th	10th	15th	20th	Reorder	1st	5th	10th	15th	20th	Reorder	Tokens	$1/\tau$	Latency	Speedup
Retrieval-based Methods																
BM25	53.7	49.3	47.9	49.9	46.9	50.3	50.9	44.9	44.1	42.9	43.2	46.0	1,545	1.9x	2.1	1.9x
Gzip	64.6	63.8	60.5	58.3	57.3	64.4	61.9	55.7	52.7	50.8	50.9	59.3	1,567	1.9x	2.1	1.9x
SBERT	72.5	67.9	63.3	65.0	66.2	68.7	65.8	57.5	54.9	53.4	55.7	61.4	1,549	1.9x	2.2	1.9x
OpenAI	73.0	65.6	66.5	65.4	65.5	69.9	65.9	57.5	56.2	54.2	55.7	61.7	1,550	1.9x	4.9	0.8x
LongLLMLingua r_k	73.9	67.7	68.7	66.0	65.6	74.3	68.5	59.1	56.8	55.3	56.9	65.2	1,548	1.9x	2.3	1.8x
Compression-based Methods																
Selective-Context	45.4	39.0	33.8	33.5	41.5	-	53.2	26.3	25.4	24.2	33.3	-	1,478	2.0x	7.4	0.6x
LLMLingua	39.7	39.5	40.4	37.1	42.3	41.5	38.7	37.3	35.7	34.1	37.5	37.1	1,410	2.1x	2.8	1.5x
LongLLMLingua	**77.2**	**72.9**	**70.8**	**70.5**	**70.6**	**76.2**	**68.7**	**59.4**	**57.3**	**55.9**	**58.4**	**66.1**	1,429	2.1x	2.9	1.4x
Original Prompt	75.7	57.3	54.1	55.4	63.1	-	68.6	57.4	55.3	52.5	55.0	-	2,946	-	4.1	-
Zero-shot	56.1						35.0						15	196x	1.1	3.7x

Table 4: Performance of different methods under different compression ratios on LongBench (Bai et al., 2023) and ZeroSCROLLS (Shaham et al., 2023) using GPT-3.5-Turbo. Considering the dataset structure, we do not use the reordering strategy here.

Methods	**LongBench**										**ZeroSCROLLS**			
	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Code	AVG	Tokens	$1/\tau$	Latency	AVG	Tokens	$1/\tau$	Latency
3,000 tokens constraint														
Retrieval-based Methods														
BM25	32.3	34.3	25.3	57.9	45.1	48.9	40.6	3,417	3x	7.5(2.1x)	19.8	3,379	3x	5.5(2.2x)
SBERT	35.3	37.4	26.7	63.4	51.0	34.5	41.4	3,399	3x	7.7(2.0x)	24.0	3,340	3x	5.9(2.1x)
OpenAI	34.5	38.6	26.8	63.4	49.6	37.6	41.7	3,421	3x	13.3(1.2x)	22.4	3,362	3x	11.7(1.0x)
LongLLMLingua r_k	37.6	42.9	26.9	68.2	49.9	53.4	46.5	3,424	3x	8.2(1.9x)	29.3	3,350	3x	6.2(2.0x)
Compression-based Methods														
Selective-Context	23.3	39.2	25.0	23.8	27.5	53.1	32.0	3,328	3x	50.6(0.3x)	20.7	3,460	3x	54.2(0.2x)
LLMLingua	31.8	37.5	26.2	67.2	8.3	53.2	37.4	3,421	3x	9.2(1.7x)	30.7	3,366	3x	7.4(1.7x)
LongLLMLingua	**40.7**	**46.2**	**27.2**	**70.6**	**53.0**	**55.2**	**48.8**	3,283	3x	10.0(1.6x)	**32.8**	3,412	3x	8.2(1.5x)
2,000 tokens constraint														
Retrieval-based Methods														
BM25	30.1	29.4	21.2	19.5	12.4	29.1	23.6	1,985	5x	4.6(3.4x)	20.1	1,799	5x	3.8(3.2x)
SBERT	33.8	35.9	25.9	23.5	18.0	17.8	25.8	1,947	5x	4.8(3.4x)	20.5	1,773	6x	4.1(3.0x)
OpenAI	34.3	36.3	24.7	32.4	26.3	24.8	29.8	1,991	5x	10.4(1.5x)	20.6	1,784	5x	9.9(1.2x)
LongLLMLingua r_k	37.8	41.7	26.9	66.3	53.0	52.4	46.3	1,960	5x	4.7(3.3x)	24.9	1,771	6x	4.7(2.6x)
Compression-based Methods														
Selective-Context	16.2	34.8	24.4	15.7	8.4	49.2	24.8	1,925	5x	47.1(0.3x)	19.4	1,865	5x	47.5(0.3x)
LLMLingua	22.4	32.1	24.5	61.2	10.4	**56.8**	34.6	1,950	5x	5.9(2.6x)	27.2	1,862	5x	4.8(2.5x)
LongLLMLingua	**39.0**	**42.2**	**27.4**	**69.3**	**53.8**	56.6	**48.0**	1,809	6x	6.1(2.6x)	**32.5**	1,753	6x	5.2(2.3x)
Original Prompt	39.7	38.7	26.5	67.0	37.8	54.2	44.0	10,295	-	15.6	32.5	9,788	-	12.2
Zero-shot	15.6	31.3	15.6	40.7	1.6	36.2	23.5	214	48x	1.6(9.8x)	10.8	32	306x	1.0(12.2x)

in LooGLE, Table 8). On the contrary, reordering can effectively alleviate the “lost in the middle” issue, thereby improving LLMs performance. 7) Using GPT2-small reduces the capture of effective tokens, but it can still achieve results close to or even slightly better than the original prompt.

E.2 MULTI-DOCUMENT QA, LONGBENCH AND ZEROSCROLLS

Table 3 and 4 present the results of different methods under various compression ratios in Multi-document QA, LongBench, and ZeroSCROLLS.

E.3 ABLATION STUDY IN MULTI-DOCUMENT QA

---

[17] https://python.langchain.com/docs/modules/data_connection/document_transformers/post_retrieval/long_context_reorder

To evaluate the contributions of different components in LongLLMLingua, we introduce following variants of it for ablation study. (1) Variants about Question-aware Coarse-grained Compression, include: ours w/o Question-awareness, which calculates question-text relevance r_k using information entropy in LLMLingua, ours w/ SBERT, which employs SBERT to compute r_k, ours w/ $p(\mathbf{x}_k^{\text{doc}}|x_i^{\text{que,restrict}})$, which replace $p(x_i^{\text{que,restrict}}|\mathbf{x}_k^{\text{doc}})$ with $p(\mathbf{x}_k^{\text{doc}}|x_i^{\text{que,restrict}})$ in Eq. 1, and ours w/o restrict, which only calculates the conditional probability corresponding to x^{que}. (2) Ours w/o Question-aware Fine-grained, which disregards Eq. (2) and only applies Iterative Token-level Prompt Compression as LLMLingua. (3) Ours w/o Dynamic Compression Ratio, where all documents share the same compression ratio in fine-grained compression. (4) Ours w/o and (5) LLMLingua w/ Subsequence Recovery, which either removes or adds the post-processing subsequence recovery strategy. (6) Ours w/ GPT2-small, which uses the GPT2-small model as the small language model.

Table 5: Ablation study on NaturalQuestions with 2x constraint using GPT-3.5-Turbo.

	1st	5th	10th	15th	20th
LongLLMLingua	**77.2**	**72.9**	**70.8**	**70.5**	**70.6**
Question-aware Coarse-grained					
- w/o Question-awareness	42.1	40.3	39.7	40.1	40.3
- w/ SBERT	73.2	68.5	65.7	66.1	66.7
- w/ $p(\mathbf{x}_k^{\text{doc}}\|x_i^{\text{que,restrict}})$	56.0	52.6	53.4	51.6	51.1
- w/o restrict	75.1	72.2	70.3	70.3	70.2
- w/o Question-aware Fine-grained	75.8	71.0	68.9	68.4	69.3
- w/o Dynamic Compression Ratio	74.4	70.7	68.7	67.9	68.1
- w/o Subsequence Recovery	76.7	71.7	69.4	69.3	69.7
- w/ Document Reordering	76.2	76.2	76.2	76.2	76.2
- w/ GPT2-small	74.6	71.7	70.1	69.8	68.5
LLMLingua	39.7	39.5	40.4	37.1	42.3
- w/ Subsequence Recovery	43.8	44.1	43.5	43.3	44.4

Table 5 shows the results of the ablation study. In summary, removing any component proposed for LongLLMLingua will lead to a performance drop regardless of the position of the ground-truth answer. This well validates the necessity and effectiveness of the proposed question-aware mechanism during coarse-to-fine compression, the dynamic compression ratio, and the subsequence recovery strategy. It also shows that applying SBERT for coarse-grained compression will result in inferior performance, which implies the superiority of our question-aware importance metric in Eq. 1 over SBERT. In addition, replacing $p(x_i^{\text{que,restrict}}|\mathbf{x}_k^{\text{doc}})$ with $p(\mathbf{x}_k^{\text{doc}}|x_i^{\text{que,restrict}})$ can greatly affect performance due to the large noise in calculating $p(\mathbf{x}_k^{\text{doc}})$ since the perplexity of document depends on many other information besides the question. Removing the restrictive statement can increase the hallucination of small language models, leading to a decrease in performance. Moreover, our subsequence recovery strategy can also bring performance gains for LLMLingua. However, without our question-aware mechanism, results from LLMLingua are still less satisfactory. For more detailed cases, please go to Appendix H.

E.4 LONGBENCH USING LONGCHAT-13B-16K

Table 6: Performance of different methods under different compression ratios on LongBench (Bai et al., 2023) using LongChat-13b. Considering the dataset structure, we do not use the reordering strategy here.

Methods	SingleDoc	MultiDoc	Summ.	FewShot	Synth.	Code	AVG	Tokens	$1/\tau$
Original Prompt	27.4	30.3	20.3	49.9	12.5	42.5	30.5	10,295	-
Retrieval-based Methods									
BM25	2.4	2.6	16.4	8.7	0.0	44.7	12.5	1,985	5x
SBERT	11.6	13.7	21.1	16.2	7.5	30.0	16.7	1,947	5x
LongLLMLingua r_k	30.3	32.4	24.5	41.0	27.5	38.1	32.3	1,960	5x
Compression-based Methods									
Selective-Context	16.1	23.5	21.8	21.4	2.5	35.9	20.2	1,925	5x
LLMLingua	20.6	22.3	22.4	35.6	0.0	35.4	22.7	1,950	5x
LongLLMLingua	31.1	34.1	24.5	45.7	28.0	48.6	35.3	1,809	6x

Table 6 presents the experiment results in the LongBench long context benchmark using LongChat-13b-16k. It can be seen that the compressed prompt can also achieve good results on other LLMs, such as LongChat-13b-16k. Specifically, 1) there is a maximum improvement of 15.5 points in synthetic tasks. Except for a slight drop in few-shot Learning, there is an improvement of 3-5 points in other tasks. 2) The performance trends of retrieval-based and compressed-based baselines are similar to the results in GPT-3.5-Turbo.

E.5 ZEROSCROLLS BREAKDOWNS

Table 7: Performance breakdown of different methods under different compression ratios on ZeroSCROLLS (Shaham et al., 2023) using GPT-3.5-Turbo.

Methods	GvRp	SSFD	QMsm	SQAL	QALT	Nrtv	Qspr	MuSQ	SpDg	BkSS	AVG	Tokens	$1/\tau$
3,000 tokens constraint													
Retrieval-based Methods													
BM25	9.7	3.4	11.7	14.3	57.1	5.9	25.7	11.2	29.6	29.6	19.8	3,379	3x
SBERT	16.5	9.8	12.3	15.2	60.0	14.6	23.4	12.1	39.4	36.4	24.0	3,340	3x
OpenAI	14.3	8.3	12.0	15.3	66.7	13.3	24.3	11.7	31.2	26.4	22.4	3,362	3x
LongLLMLingua r_k	19.5	11.6	14.7	15.5	66.7	20.5	27.6	13.0	60.8	43.4	29.3	3,350	3x
Compression-based Methods													
Selective-Context	20.8	9.1	11.7	13.4	50.0	9.8	26.1	11.0	46.0	9.5	20.7	3,460	3x
LLMLingua	18.7	10.0	14.9	16.8	61.9	26.9	27.2	23.4	62.9	44.5	30.7	3,366	3x
LongLLMLingua	21.9	12.7	15.5	17.0	66.9	27.6	31.1	23.8	65.6	46.4	32.8	3,412	3x
2,000 tokens constraint													
Retrieval-based Methods													
BM25	8.8	2.5	11.1	13.5	60.0	7.0	4.9	20.3	39.9	32.9	20.1	1,799	5x
SBERT	10.2	7.9	13.7	13.2	60.0	8.1	10.8	1.7	37.2	42.8	20.5	1,773	6x
OpenAI	11.1	8.0	11.8	13.6	60.0	7.1	13.2	4.0	33.6	43.6	20.6	1,784	5x
LongLLMLingua r_k	18.2	9.8	12.3	15.9	57.1	10.1	17.8	7.3	57.7	42.3	24.9	1,771	6x
Compression-based Methods													
Selective-Context	19.0	8.4	9.7	12.4	47.0	12.5	21.6	11.5	41.2	11.0	19.4	1,865	5x
LLMLingua	19.4	11.9	13.1	16.0	62.1	23.7	24.0	22.4	33.9	44.9	27.2	1,862	5x
LongLLMLingua	19.9	12.3	14.7	16.5	64.9	27.4	30.6	23.5	68.3	47.1	32.5	1,809	6x
Original Prompt	21.8	12.1	17.9	17.4	66.7	25.3	29.8	20.0	69.7	44.1	32.5	9,788	-
Zero-shot	9.4	3.0	8.6	11.4	42.9	10.6	12.4	5.5	4.2	0.0	12.8	32	306x

Table 7 presents a detailed performance breakdown on the ZeroSCROLLS benchmark. It can be observed that in the four summarization tasks - GvRp, SSFD, QMsm, SQAL, LongLLMLingua closely matches or slightly surpasses the original results under two compression constraints. Meanwhile, in the four long context QA tasks - Qsqr, Nrtv, QALT, MuSQ, there is a significant improvement. Notably, in the MuSiQue task, which is based on a question-answering dataset from books and movie scripts, there is a 2.1 point increase even under a 2,000 tokens constraint. It's worth mentioning that MuSiQue is a multi-hop question-answering dataset that requires LLMs to utilize global information for long dependency QA. LongLLMLingua can also improve by 3.5 points under a 6x compression ratio. In the two ordering tasks, SpDg and BkSS, LongLLMLingua can better retain globally sensitive information, resulting in a 3.0 point improvement in BkSS after prompt compression.

It's important to note that although the ZeroScrolls validation dataset is relatively small, it still demonstrates conclusions similar to previous experimental observations across various methods and tasks. Furthermore, this study conducted an in-depth analysis of the multi-hop QA task - MuSiQue, and another long context benchmark - LooGLE. The results can be found in Appendix E.7 and Appendix E.6.

E.6 LOOGLE

Table 8 presents the experiment results in the LooGLE long dependency benchmark, which features longer prompts (∼30k) and more global dependencies. From the table, we can observe that: 1) LongLLMLingua can effectively improve the performance of long context tasks by compressing prompts, even for long dependency tasks. The results show that LongLLMLingua significantly improves performance in tasks such as retrieval, timeline reorder, and computation, with the maximum improvement reaching 15.9 points. 2) The document reorder in LongLLMLingua is effective in all types of tasks, even in tasks highly related to the timeline, it can effectively improve performance by alleviating the "lost in the middle" issue. 3) Retrieval-based methods tend to lose performance in tasks that have longer dependencies, such as computation and reasoning. 4) For compression-

Table 8: Performance of different methods on LooGLE (Li et al., 2023b) long dependency QA.

Methods	Retrieval	Timeline Reorder	Computation	Reasoning	AVG	Tokens	$1/\tau$
Retrieval-based Methods							
BM25	20.4	21.7	8.2	26.3	19.2	3,185	10x
SBERT	28.9	21.1	10.7	27.2	22.0	3,169	10x
LongLLMLingua r_k	38.6	32.2	16.2	26.3	28.3	3,158	10x
Compression-based Methods							
Selective-Context	16.7	5.0	2.3	17.6	10.4	3,710	8x
LLMLingua	10.0	25.0	13.3	21.1	17.3	3,404	9x
LongLLMLingua	**40.0**	**35.0**	**19.7**	**33.6**	**32.1**	3,121	10x
LongLLMLingua w/o Reorder	39.3	33.8	18.7	31.6	30.9	3,119	10x
Original Prompt	24.1	20.9	13.5	32.1	22.6	30,546	-
Zero-shot	8.7	6.3	1.2	14.5	7.7	43	710x

based methods, due to the difficulty in perceiving question information, there tends to be a larger performance loss in retrieval tasks within long contexts.

E.7 MUSIQUE

Table 9 presents the results from the MuSiQue multi-hop question-answer dataset. From the table, it can be observed that in the multi-hop QA task, requiring global information: 1) LongLLMLingua can reduce noise in the prompt by eliminating irrelevant information and putting more related information at the beginning or end of the prompt, thereby improving performance by 5.4 points. 2) The performance drop is more pronounced for retrieval-based methods, particularly for n-gram-based methods like BM25. Due to long dependencies, direct matching information is lost, resulting in less relevant information being recalled. 3) The performance of compression-based methods is slightly different. Selective-Context does not distinguish between different modules' sensitivity, resulting in a loss of question and instruction-related information, thereby leading to poorer performance. However, LLMLingua can still retain relevant key information at around a 2x compression ratio. 4) The ablation experiments show that every module designed in LongLLMLingua plays a role in the multi-hop task. The removal of the question-aware coarse-grained and w/ $p(\mathbf{x}_k^{\text{doc}}|x_i^{\text{que,restrict}})$ modules, which have difficulty in perceiving the importance distribution of corresponding questions, can cause a drop of up to 8 points. Removing the restrict prompt in the question-aware coarse module can also cause a 2-point drop due to the hallucination issue of small LLM. In addition, removing question-aware fine-grained, dynamic compression ratio, and document reordering can all cause a drop of 0.5-2.8 points. 5) Moreover, if the small language model in LongLLMLingua is replaced with GPT2-small, it can further improve the acceleration ratio and still achieve a result that is 2.6 points better than the original prompt.

Table 9: Performance of different methods and ablation study on MuSicQue (Trivedi et al., 2022) with 2x constraint using GPT-3.5-Turbo.

Methods	F1	Tokens	$1/\tau$
Original Prompt	45.8	2,427	-
BM25	28.5	1,295	1.9x
SBERT	36.2	1,288	1.9x
LongLLMLingua r_k	46.3	1,295	1.9x
Selective-Context	19.6	1,141	2.1x
LLMLingua	40.1	1,110	2.2x
LongLLMLingua	**51.2**	1,077	2.3x
Question-aware Coarse-grained			
- w/o Question-awareness	43.2	1,076	2.3x
- w/ SBERT	47.3	1,070	2.3x
- w/ $p(\mathbf{x}_k^{\text{doc}}|x_i^{\text{que,restrict}})$	44.0	1,066	2.3x
- w/o restrict	49.2	1,078	2.3x
- w/o Question-aware Fine-grained	48.4	1,118	2.2x
- w/o Dynamic Compression Ratio	48.2	1,090	2.2x
- w/o Subsequence Recovery	50.7	1,077	2.3x
- w/o Document Reordering	49.2	1,077	2.3x
- w/ GPT2-small	48.4	1,095	2.2x

E.8 LATENCY EVALUATION

We conducte end-to-end latency testing on a V100-32G, using the prompts from Multi-document QA, LongBench, and ZeroSCROLLS in the API call, and results are shown in Table 1, 3and 4. The latency includes the time cost for prompt compression and the request time for LLMs, with multiple measurements taken and averaged over. Results demonstrate that LongLLMLingua does indeed speed up the overall inference under different compression ratios and scenarios. Moreover, with the compression ratio increasing, the acceleration effect becomes more pronounced up to 2.6x. However, the OpenAI embedding and Selective-Context results in longer latency time, due to repeated API calls and the sequential entropy calculation of semantic units, respectively.

F ECONOMIC COST

Table 10: The inference costs(per 1,000 samples $) for various datasets using GPT-3.5-Turbo.

	Multi-document QA	LongBench	ZeroScolls	MuSicQue	LooGLE
Original	4.6	31.5	30.6	3.8	93.6
Ours	1.3	3.0	3.2	1.8	5.6

Table 10 presents the estimated per 1,000 samples inference costs for various datasets, encompassing input prompts and generated output text, based on GPT-3.5-Turbo pricing[18]. Our approach demonstrates substantial savings in computational resources and monetary expenses, particularly in long context situations. Cost reductions of $3.3, $28.5, $27.4, $2.0, and $88.0 per 1,000 samples are observed for Multi-document QA, LongBench, ZeroScrolls, MuSiQue, and LooGLE, respectively.

G RELATED WORKS

Long context for LLMs. Recent research has focused on expanding the window size of LLMs. Main approaches include: (1) Staged pre-training (Nijkamp et al., 2023) which gradually increases the context window; (2) Modifying (Press et al., 2022) or interpolating position embeddings (Chen et al., 2023; Peng et al., 2024; Han et al., 2023); (3) Using linear or sparse attention mechanisms (Ding et al., 2023; Sun et al., 2023); (4) Utilizing external memory modules for context storage (Bertsch et al., 2023; Tworkowski et al., 2023). While these methods address context window expansion, their impact on downstream task performance has yet to be discussed.

Information distribution in prompt. Recent empirical experiments have shown that LLM performance decreases with less effective information in a prompt (Bai et al., 2023; Li et al., 2023a; Shi et al., 2023). Moreover, the position of relevant information in a prompt has a significant impact on performance(Wu et al., 2023). Liu et al. (2024) suggests that LLMs have more difficulty comprehending information located in the middle of a prompt compared to those at the edges.

Retrieval methods can be categorized as dense or sparse retrieval methods. Sparse retrieval methods, like BM25, determine the relevance between queries and documents based on n-gram information. Conversely, dense retrieval methods assess the relevance between queries and documents in latent space using dense vectors, such as SentenceBERT (Reimers & Gurevych, 2019) and OpenAI Embedding. Recently, Jiang et al. (2023b)) proposed an unsupervised dense retrieval method that leverages traditional compression algorithms, such as gzip, and k-nearest neighbors.

Prompt compression methods can be grouped into three main categories: (1) Token pruning (Goyal et al., 2020; Kim & Cho, 2021; Modarressi et al., 2022) and token merging (Bolya et al., 2023), which need model fine-tuning or intermediate results during inference and have been used with BERT-scale models. (2) Soft prompt tuning methods like GIST (Mu et al., 2023), AutoCompressor (Chevalier et al., 2023), and ICAE (Ge et al., 2024), which require LLMs' parameter fine-tuning, making them suitable for specific domains but not directly applicable to black-box LLMs. (3) Information-entropy-based approaches such as Selective Context (Li et al., 2023c) and LLMLingua (Jiang et al., 2023a), which use a small language model to calculate the self-information or perplexity of each token in the original prompt and then remove tokens with lower perplexities.

H ABLATION ANALYSIS

Figure 6 illustrates the compressed prompts from the Multi-document QA dataset, comparing the use of contrastive perplexity at a high compression ratio (30x). It shows that without question-aware

---

[18] https://openai.com/pricing

token-level prompt compression, LongLLMLingua tends to compress key information, a tendency that becomes more pronounced at higher compression ratios. Conversely, employing contrastive perplexity allows for better detection of key information related to the question within the context, thus preserving key information within the compressed prompt.

Ours w/o Token-level Question-aware:
Compressed Prompt:
Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).
Document [1](: Physics)gen,, who received2K, which is ,73,0 in0. Johnen only to twice6. Mariaie won, for.g was, until1estate he. Two:Mayer (1963). As of 2017, the prize has been awarded
Question: who got the first nobel prize in physics
Answer:
LLMs' Response:
No answer found in the given search results.

Ours w/ Token-level Question-aware:
Compressed Prompt:
Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).
1Title: List of Nobelates in The first Nobel Prize was1 to **Wilhelmrad**, of who received 1582 which,70 in0 en the prize. Skska also won two Nobeles for physics3g01, theate he women prize:ertMayer (1963). As of 2017, the prize has been awarded
Question: who got the first nobel prize in physics
Answer:
LLMs' Response:
Wilhelmrad
LLMs' Response after Subsquence Recovery:
Wilhelm Conrad Röntgen
Ground Truth:
Wilhelm Conrad Röntgen

Figure 6: Comparing the compressed prompt and LLMs' response before and after using Question-aware Fine-grained Compression and Subsequence Recovery($1/\tau$ = 30x, high compression ratio setting) from NaturalQuestions Multi-document QA (Liu et al., 2024) using GPT-3.5-Turbo.

I CASES STUDY

Figures 7, 8, and 9 display the outcomes before and after compression, as well as the LLMs' responses in various scenarios.

Original Prompt:

...

Document [1](Title: Dancing on Ice) It was confirmed on 25 January 2018, that Dancing on Ice had been recommissioned for an eleventh series to air in $\boxed{\textbf{2019}}$.

...

Compressed Prompt:

Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).

1Title: Dancing on was confirmed on 2 January 2018 that Dancing on had been recommissioned for an eleventh series air in $\boxed{\textbf{209}}$.

Document [2Title: Dan on) Dan on Ice Dancing on British presented by Phillip Schof alongside Holly Willough from 26 to 2011, and Christine Bleakley from 2012 to 204 The show consists of celebrit and professional partners figure skating in front of a panel of judges The, broadcast on ITV, started on January 2006 and ended on 9 March 2014 after showćontract not renewed by ITV On 4 September 2017, it was announced that rev series would on I 7 January 201 Sch and Willby returning as a

5(: on (on () The third series of a from January to168TV. The from Saturdays, with Holby present Kar,y Sliner Robin Cins returned to Panel", with Ruth H joining the panel as replacement for Natalia Bestova. The commission of the was confirmed by at the07 announcedova depart the series Robinen Bar,ater and Jasoniner announced

7(on ( )) Dan 2 second of Dan on a from January to1207 ITV It presented Phillip Sch Holly Willough, and judged the "I P consisting Nicky Slater, Nataliaian Karenres Jason Gardiner Karen Barber and Robin Cousins Jaynevill and Christopher Dean co and trained the contestants In this series, cele to ten in first series. The series was won former Kyran Bracken, with Mel Lambert the winner. It announced thatenresge

Document [](on Ice on 08 on TV edition started 8 TV2 The Russian version "анду) being on channel0, and renamed in8 to " Ice" (). Its counterpart called "Ice Age (, "Stars on Ice on Channel Oneak IceHviezdyǏJ. The Turkish version" is called Dans" ("ance on Document1 on Ice its, all,é () and Sje Chris de In series.2 edition

](: on Ice world) Dan Ice is a made competition world format, and been subsequently Italy Chile where titled after series There have a, the show was broadcast on Channel 13 as a

Document [17](Title: Dancing on Ice) the insight to the training of the celebrities over the last week. It was presented by television presenter Ben Shephard and former contestant and "Loose Women" star Coleen Nolan. The show was broadcast from 8 pm to 8.30 pm on Friday evenings on ITV throughout the duration of the main shows season. STV who broadcast the main show did not broadcast this on the Friday evening but after repeating the previous weekś main show on the following Saturday afternoon. Due to poor ratings, "Dancing on Ice Friday" was axed prior to the 2011 series. The show was based in the

Question: when is dancing on ice on the tv

Answer:

LLMs' Response:

209

LLMs' Response after Subsequence Recovery:

2019

Ground Truth:

2019

Figure 7: Cases study on NaturalQuestions Multi-document QA dataset (Liu et al., 2024) in 4x constraint using GPT-3.5-Turbo.

Compressed Prompt:
Please complete the code given below.

```
public class MessageArchiveManagement
    private static final long MILLISECONDS_IN_DAY = 24 * 00 *0;
    public static final long_CUP = MCON_DAY
    /.../
          .("",.getStart
          add
 ifget() >0
           Node end("
            end.("
            endNode.Value("", Util.getTimestamp(query.getEnd
addNode
        }        if (.withid null && contact null && !isference
           Node with("           .with
           .Value("valuewith
           .(
        //    queryMessageive(connection, nextQuery
            final(connectionProtocol(), query
            synchronized (eries)
            //    queries.add(nextQuery } }
    public boolean queryInProgress( contact, OnLoaded
    moreMessagesLoadedListener)
       ized (eries)
            (Query query : queries)
                if(query.getWith().equals(contact.getUserId()))
    if (query.onMoreMessagesLoaded == null &&MessagesListener
    null) query.setOnMoreMessagesLoaded(Listener}
                    return true;}} return false;}}
    private void finalizeQuery(Protocol protocol, Query query) {
        synchronized (queries) {
            .remove(query); }
        Contact contact = null;
        if (query.getWith() != null) {
            contact = protocol.getItemByUID(query.getWith()); }
        if (contact != null) {
```

Next line of code:
LLMs' Response:

```
        contact.setLastMessageTransmitted(query.getEnd());\n
```

Ground Truth:

```
        if (contact.setLastMessageTransmitted(query.getEnd())) {
```

Zero-shot LLMs' Response:

```
        contact.removeQuery(query);\n
```

Figure 8: Cases study on lcc code completion task in LongBench benchmark (Bai et al., 2023) in 2,000 constraint using GPT-3.5-Turbo.

Compressed Prompt:
Please determine the Type of the question below. Here are some examples of questions.
Question: How is energy created ? Type Manner of an action
Question: What is chocolate ? Type: Definition of something
Question: What is a bone marrow transplant ? Type: Definition of something
Question: What is fear of odors , body , ? Type Disease and medicine
Question: What was the Vietnam War ? Type: Definition of something
Question: was education system in 16s ? Type: Other entity
Question: What is IP address ? Type: Definition of something
Question: are the differences in Catholic Methodist religions ? Type of something
...
Question: When was San fire ? : Date
Question: CNN began broadcasting in what year ? Type: Date
Type: Manner of an action
Question: What the l behind the ir in the eye called ? Type Equ term
Type: Date
Question: What the former name of Zimbabwe ? Type: termType something
Question: What is troilism ? Type: Definition of something
: What is origin of the word , Type: of something
: do you name to social security number ? Type Manner of an action
: that of an employee Universal and Export ? Type Individual
: anesthetic did Queen Victoria allow to be for the birth of her seventh , in 183 ? Type: Disease and medicine
: Where isyer 's rock ? Type location
Question: What isymnophobia ? Type: Definition of something
...
Type burns the most calories ?
Type Sport
: In what book I find story of Aladdin ? Type In, book and piece an have sex ?
Type: Manner of an action: What is the acron for rating forer ?
Type Abbreviation
: are the Baltic States ? Type: Definition of something
: What is appearance , that violates the standards of sexual mor ? Type
: Where did the May people live ? : location
: What population Kansas ? Type number
: was the hurr ? Type: Event
: 's a score aymnast exercise ? Type: number
: year become a state ? Type: Date
do go school ? Type Reason
...
Question: What is a fuel cell ?
Type:
LLMs' Response:
Definition of something
LLMs' Response after Subsquence Recovery:
Definition of something
Ground Truth:
Definition of something

Figure 9: Cases study on trec few-show learning in LongBench benchmark (Bai et al., 2023) in 2,000 constraint using GPT-3.5-Turbo.