Pause-Tuning for Long-Context Comprehension: A Lightweight Approach to LLM Attention Recalibration

James Begin [1] **Namit Agrawal** [1] **Eshan Singh** [1] **Yicheng Fu** [1] **Sean O'Brien** [1] **Vasu Sharma** [1] **Kevin Zhu** [1]

Abstract

LLMs have demonstrated remarkable proficiency in understanding tasks but continue to struggle with long-context comprehension, particularly with content located in the middle of extensive inputs. This limitation, known as the Lost-in-the-Middle (LITM) problem, hinders models from fully processing and utilizing information across lengthy contexts. To address this issue, we introduce pause-tuning, a technique that redistributes attention to enhance comprehension of long-context inputs. Our approach fine-tunes language models on datasets with inserted pause tokens, segmenting inputs into manageable parts. We evaluate pause-tuning against alternative approaches using the Needle-in-a-Haystack (NIAH) and LongBench v2 benchmarks, in which models must retrieve specific information and answer challenging multiple-choice questions respectively based on long contexts. Experimental results demonstrate significant performance gains, suggesting that pause-tuning successfully enhances attention redistribution and improves long-context retention. We also observe significant changes in the attention distribution when pause-tuning is applied. The code and data are available at https://anonymous.4open.science/r/LITM-PauseTokens-7357.

1. Introduction

Language models like GPT (Brown et al., 2020) and LLaMA (Grattafiori et al., 2024) have demonstrated remarkable utility in tasks such as summarization, long document analysis, and contextual understanding (Minaee et al., 2024). Effectively handling long contexts is essential for maintaining the accuracy and reliability of a model's output in these applications. However, language models often suffer from the lost-in-the-middle problem (Liu et al., 2023), where they disproportionately focus on the beginning and end of sequences while neglecting information in the middle. Existing attempts at solutions often fall short of being broadly applicable. Many approaches rely on computationally intensive mechanisms or involve modifications of the base language model other than simple fine-tuning (He et al., 2024; Tworkowski et al., 2023; Liu & Abbeel, 2023). While effective in certain scenarios, these methods may be impractical in resource-constrained environments or general-purpose applications.

To address this gap, we propose a novel approach that utilizes pause tokens (Goyal et al., 2024) to mitigate the LITM problem. Pause tokens are markers that are strategically inserted into the input sequence, intended to recalibrate the model's attention distribution. These tokens prompt the model to pause and process additional computation before proceeding with the rest of the sequence. By segmenting the input sequence into smaller, more manageable chunks, pause tokens allow the model to process each segment with greater focus and parity. This simple yet effective method offers a lightweight alternative to existing resource-intensive techniques. We investigate various strategies for inserting pause tokens, as depicted in Figure 1.

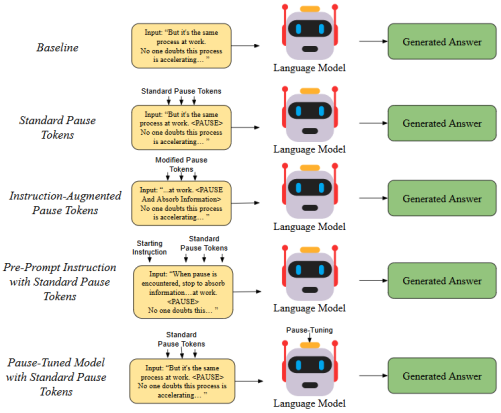

Figure 1. We propose five potential techniques for pause token injection and test them on the needle-in-a-haystack and LongBench v2 evaluation frameworks.

[1]Algoverse AI Research. Correspondence to: Kevin Zhu <kevin@algoverse.us>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

We experimentally evaluate pause token efficacy by comparing base models with those fine-tuned for pause-aware long-context processing. Our results demonstrate that:

- Pause-tuning consistently improves long-context retention and processing, outperforming alternative techniques.
- Pause tokens induce meaningful shifts in attention distribution, enhancing information retrieval across extended input sequences.

Our findings highlight pause-tuning as an effective and computationally efficient mechanism for mitigating long-context deficiencies in LLMs.

2. Related Work

2.1. Lost-in-the-Middle

Large language models exhibit a U-shaped performance curve when processing long inputs, demonstrating a pronounced primacy and recency bias (Liu et al., 2023). That is, they allocate greater attention to the beginning and end of a sequence while neglecting the middle (Khandelwal et al., 2018; Press et al., 2021). Xiao et al. (2023) further observed that models assign disproportionately high attention scores to initial tokens, even when these tokens lack semantic significance. This phenomenon extends to multi-document question-answering tasks as well as key-value retrieval tasks, both of which are closely related to our evaluation methods. Building on these findings, our research investigates whether similar biases emerge in our specific context and examines their implications.

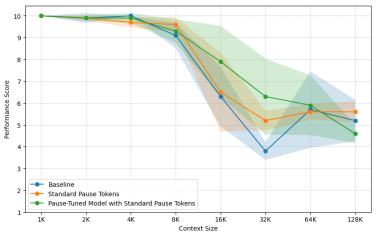

Figure 2. LLaMA 3.2 3B Instruct needle-in-a-haystack performance across techniques. The score declines as context length increases. Pause-tuning consistently outperforms other methods, except at the 128K length.

2.2. Pause Tokens

Goyal et al. (2024) introduced the concept of pause tokens in language model training. Their approach involves inserting learnable pause tokens during pretraining and finetuning, showing improvements on various Question-Answer tasks (Kwiatkowski et al., 2019; Talmor et al., 2019; Rajpurkar et al., 2016). Expanding on this idea, Rawte et al. (2024) proposed the "Sorry, Come Again" (SCA) prompting technique, which integrates optimal paraphrasing with pause token injection. This method has been shown to effectively mitigate hallucinations in large language models, further underscoring the potential of pause tokens in enhancing model reliability and interpretability.

3. Method

3.1. Token Placement Strategy

We employ a systematic approach to injecting pause tokens across different experimental configurations. The most straightforward implementation involves inserting a special "<PAUSE>" token after each paragraph in the testing context. This token serves as a standardized pause marker, facilitating natural segmentation within the input sequence. By introducing these structured breaks, our approach enables the model to redistribute attention more effectively, ensuring comprehensive processing of information from all parts of the input.

3.2. Experimental Configurations

Our study explores various approaches for pause token insertion, as illustrated in Figure 1, using trials without input sequence modifications as the baseline for comparison. We evaluate the following techniques across 15 context depths and 3 trials for single needle tests and 15 randomized trials for multi-needle tests to identify the optimal method:

1. Standard Pause Tokens: Standard pause tokens are inserted after every paragraph in the input sequence.
2. Instruction-Augmented Pause Tokens: Pause tokens, followed by an explicit instruction to "stop and absorb the information [the model] has just read," are inserted after every paragraph in the input sequence.
3. Pre-Prompt Instruction with Standard Pause Tokens: A general instruction at the beginning of the prompt directs the model to stop and absorb information after every pause token, while standard pause tokens are inserted after every paragraph in the input sequence.
4. Pause-Tuned Model with Standard Pause Tokens: Standard pause tokens are inserted after every paragraph in the input sequence, which is then processed by a model fine-tuned with standard pause tokens in long contexts.

The use of these variations is intended to systematically determine the optimal method for enhancing the attention mechanism in long-context tasks. Technique 2 and Technique 3 evaluate whether the model requires additional in-

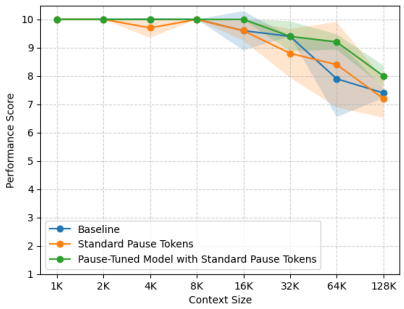

Figure 3. LLaMA 3.1 8B Instruct needle-in-a-haystack performance across techniques. The score declines as context length increases. Pause-tuning consistently outperforms other methods at all context lengths.

structions or can naturally interpret pause tokens. Technique 4 employs pause-tuning, aligning the model with the pause token structure to enhance performance.

4. Experiments

4.1. Models

We evaluate several widely used large language models (LLMs) on NIAH, including GPT-3.5 Turbo 0125 (Brown et al., 2020), GPT-4o Mini 2024-07-18 (Hurst et al., 2024), LLaMA 3.2 1B Instruct, LLaMA 3.2 3B Instruct, and LLaMA 3.1 8B Instruct (Grattafiori et al., 2024). For Technique 4, we employ pause-tuned versions of the LLaMA 3.2 3B Instruct and LLaMA 3.1 8B Instruct models. We expand our experiment to also include the instruction-tuned versions of Phi 4 mini (Abouelenin et al., 2025), Phi 4 (Abdin et al., 2024), and Gemma 3 12B (Team et al., 2025) for testing on LongBench v2.

4.2. Evaluation

We conduct our first set of experiments on the Needle-in-a-Haystack evaluation framework (Kamradt, 2023). In this framework, a critical piece of information (the "needle") is embedded within a lengthy context (the "haystack"). Our evaluation spans various context lengths, ranging from 1K to 128K tokens, depending on each model's input capacity. Additionally, we assess both single-needle and multi-needle scenarios to examine the effectiveness of our method across different conditions. For multi-needle inputs, we embed three distinct needles within the context.

The results from each trial are assigned a score from 1 to 10 based on the success of the information retrieval attempt, as described in Appendix A.

We further conduct experiments using the LongBench v2 benchmark (Bai et al., 2025). LongBench v2 is designed to assess LLM performance on long-context multiple choice problems requiring both understanding and reasoning across various tasks. Our tests span a range of 1K to 128K tokens, across six categories: single-document QA, multi-document QA, long in-context learning, long-dialogue history understanding, code repository understanding, and long structured data understanding.

4.3. Datasets

We use the Deep Essays Dataset (Gibin & Acharya, 2024) and the DAIGT Gemini-Pro 8.5K Essays dataset (Demir, 2024) for fine-tuning and a collection of Paul Graham's essays (Graham, 2001/2023) to create the haystack for testing.

4.4. Pause-Tuning

We fine-tune two LLaMA models for pause tokens in long contexts. These models were selected for their ability to handle sequences of up to 128K tokens and their strong performance in instructional tasks.

To construct an appropriate fine-tuning dataset, we concatenate multiple shorter essays and systematically inject pause tokens until the target context length is reached. Additionally, we embed a randomly selected piece of information-a "needle"-within this extended context "haystack" to assess retrieval capabilities. The models are trained using LoRA (Hu et al., 2021) and Unsloth AI (Han et al., 2023). The hyperparameters for training are in Appendix C. The training prompt adopts a one-shot format, consisting of four key components: an instruction outlining the purpose of the fine-tuning, a long-context input with injected pauses that contains the needle, an example user query that depends on retrieving the embedded information, and a response that reproduces the needle verbatim as it was originally inserted into the essay.

5. Results

Technique	1	2	3	4
GPT 3.5	0.37	0.37	0.37	—
GPT 4o	1.68	2.15	1.87	—
LLaMA 3.2 1B	2.53	3.75	11.44	—
LLaMA 3.2 3B	1.03	-1.70	-1.72	10.61
LLaMA 3.1 8B	-1.67	-2.60	0.14	3.57

Table 1: Percent Change for each technique compared to baseline, averaged across all context lengths.

5.1. Needle-in-a-Haystack

The results for the baseline and each technique using a single needle are presented in Table 5, while the results for multiple needles can be found in Appendix F. A visualization of the performance scores can be found in Figure 2

and Figure 3. Pause-tuned models demonstrates substantial improvements, with 10.61% and 3.57% increases, across both models, supporting our hypothesis that combining fine-tuning with pause tokens yields the best results. This aligns with the intuition that consistent structural training enhances performance.

5.2. LongBench v2

Table 4 and Table 2 present the results on the LongBench v2 benchmark. We observe similar trends to the NIAH testing, and pause-tuning proves to be the most successful technique again, with 6.20% improvement averaged over both models. Simpler pause token injection also delivers consistent performance gains on state-of-the-art models. These benefits are particularly pronounced in specific domains (see Appendix G), such as improving GPT-4.1's score in Code Repository Understanding by 12.5% and Claude 3.5 Sonnet's score in Long-dialogue History Understanding by 57.4%.

Technique	1	2	4
GPT 4o	4.70	-15.79	—
LLaMA 3.2 1B	2.46	0.10	—
LLaMA 3.2 3B	5.21	-5.46	6.22
LLaMA 3.1 8B	3.89	6.41	6.17
Phi 4 Mini	-0.44	-4.13	—
Phi 4	2.88	5.60	—
Gemma 3 12B	8.72	2.09	—

Table 2: Percent Change for each technique compared to baseline, averaged across all context lengths.

6. Attention Analysis

Figure 4 depicts the scaled attention distribution across different layers when generating the first answer token for the baseline and Techniques 1 and 5. Due to computational constraints, we visualize approximately 6500 token input sequences, which is a similar point to when performance starts to degrade for smaller models. The observed retrieval improvements for sequences up to 128K tokens suggest that this pattern likely extends to longer contexts.

The insertion of pause tokens significantly transforms the attention distribution. We observe distinct attention spikes at the locations of several pause tokens. These findings suggest pause tokens act as anchors, interrupting attention decay and promoting more thorough engagement across sections. The model likely treats pause-separated paragraphs as distinct sections, refreshing attention and reducing information loss. This structured attention recalibration provides insight into the improved retrieval performance observed in long contexts.

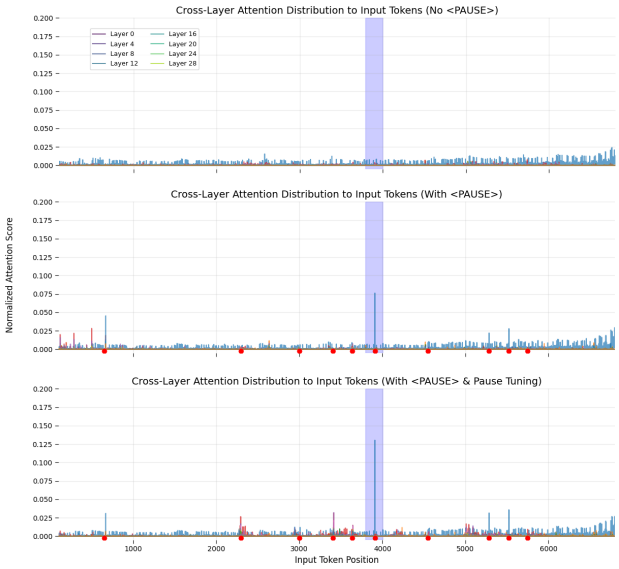

Figure 4. Attention distribution for LLaMA 3.1 8B with normalized attention scores across different layers when generating the first answer token for baseline, standard pause token insertion, and pause-tuned with pause tokens inserted. The purple highlight indicates the position of the needle. The red dots indicate inserted pause tokens.

7. Limitations

One limitation is that our study evaluates the pause-tuning technique only on relatively small models (<12B parameters). Whether these findings extend to large-scale models remains an open question and requires further investigation.

A further limitation exists in the representation and placement of pause tokens. At present, pause tokens are positioned to reflect the structure of paragraphs. However, exploring alternative strategies, such as semantic segmentation or adaptive placement based on textual complexity, could enhance the effectiveness of pause-tuning.

8. Conclusion

We introduced pause-tuning, an effective, lightweight solution for the LITM problem and long-context comprehension challenges in LLMs. By strategically injecting pause tokens into input sequences, we enhance attention redistribution, enabling models to retrieve information more effectively across extensive contexts. Our experiments demonstrate significant improvements in retrieval performance over baseline models on both benchmarks. The results confirm that pause-tuning consistently enhances long-context retention across various input lengths. These findings highlight pause-tuning's potential to mitigate LITM issues, enabling more robust long-context processing in future LLMs.

References

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Abouelenin, A., Ashfaq, A., Atkinson, A., Awadalla, H., Bach, N., Bao, J., Benhaim, A., Cai, M., Chaudhary, V., Chen, C., et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Bai, Y., Tu, S., Zhang, J., Peng, H., Wang, X., Lv, X., Cao, S., Xu, J., Hou, L., Dong, Y., Tang, J., and Li, J. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks, 2025. URL https://arxiv.org/abs/2412.15204.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., and Amodei, C. W. . D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Demir, E. Daigt gemini-pro 8.5k essays, 2024. URL https://www.kaggle.com/datasets/datafan07/daigt-gemini-pro-8-5k-essays.

Gibin, W. O. and Acharya, M. Deep essays dataset, 2024. URL https://www.kaggle.com/dsv/7607799.

Goyal, S., Ji, Z., Rawat, A. S., Menon, A. K., Kumar, S., and Nagarajan, V. Think before you speak: Training language models with pause tokens, 2024. URL https://arxiv.org/abs/2310.02226.

Graham, P. Essays. http://www.paulgraham.com/articles.html, 2001/2023. A collection of essays published online between 2001 and 2023.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., and Ma, A. H. . Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Han, D., Han, M., and team, U. Unsloth, 2023. URL http://github.com/unslothai/unsloth.

He, J., Pan, K., Dong, X., Song, Z., Liu, Y., Sun, Q., Liang, Y., Wang, H., Zhang, E., and Zhang, J. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training, 2024. URL https://arxiv.org/abs/2311.09198.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., Madry, A., Baker-Whitcomb, A., Beutel, A., Borzunov, A., Carney, A., Chow, A., Kirillov, A., Nichol, A., and Malkov, A. P. . Y. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Kamradt, G. Needleinahaystack. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Khandelwal, U., He, H., Qi, P., and Jurafsky, D. Sharp nearby, fuzzy far away: How neural language models use context. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 284–294, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1027. URL https://aclanthology.org/P18-1027/.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. 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/.

Liu, H. and Abbeel, P. Blockwise parallel transformer for large context models, 2023. URL https://arxiv.org/abs/2305.19370.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. Large language models: A survey, 2024. URL https://arxiv.org/abs/2402.06196.

Press, O., Smith, N. A., and Lewis, M. Shortformer: Better language modeling using shorter inputs. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *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. 5493–5505, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.427. URL https://aclanthology.org/2021.acl-long.427/.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Rawte, V., Tonmoy, S. M. T. I., Zaman, S. M. M., Priya, P., Chadha, A., Sheth, A. P., and Das, A. "sorry, come again?" prompting – enhancing comprehension and diminishing hallucination with [pause]-injected optimal paraphrasing, 2024. URL https://arxiv.org/abs/2403.18976.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL https://arxiv.org/abs/1811.00937.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Tworkowski, S., Staniszewski, K., Pacek, M., Wu, Y., Michalewski, H., and Miłoś, P. Focused transformer: Contrastive training for context scaling, 2023. URL https://arxiv.org/abs/2307.03170.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

A. Needle-in-a-Haystack Scoring

To score the retrievals for the needle-in-a-haystack test, the following framework was used, with relevance assessed by the model:
Score 1: The answer is completely unrelated to the reference.
Score 3: The answer has minor relevance but does not align with the reference.
Score 5: The answer has moderate relevance but contains inaccuracies.
Score 7: The answer aligns with the reference but has minor omissions.
Score 10: The answer is completely accurate and aligns perfectly with the reference.

B. Prompt Formatting

Prompt without ¡PAUSE¿ tokens

Below is an instruction that describes a task, paired with a context that provides further information. An input will request information from the context. Write a response that appropriately completes the request.

###Instruction:

You are a helpful assistant that will be provided a context which the user wants to ask a question about, your job is to answer the question with only statements provided in the context and nothing else.

###Context:

{context}

###Input:

{input}

Prompt with ¡PAUSE¿ tokens

Below is an instruction that describes a task, paired with a context that provides further information. An input will request information from the context. Write a response that appropriately completes the request.

###Instruction:

You are a helpful assistant that will be provided a context which the user wants to ask a question about, the context has ¡PAUSE¿ tokens that tell you when to take a pause to comprehend the context before continuing, your job is to answer the question with only statements provided in the context and nothing else.

###Context:

{context}

###Input:

{input}

C. Hyperparameters

LoRA Parameters	
Rank (r)	16
Alpha	16
Dropout	0
Target Modules	[”q_proj”, ”k_proj”, ”v_proj”, ”o_proj”, ”gate_proj”, ”up_proj”, ”down_proj”]
Training Parameters	
Batch Size	2
Gradient Accumulation Steps	4
Learning Rate	2e-4
Weight Decay	0.01
Warmup Steps	6
Steps	60
LR Scheduler	linear
Optimizer	adamw_8bit
Other	
Mixed Precision	fp16
Quantization Bits	4 bit

Table 3: Training Hyperparameters

D. LongBenchv2 Detailed Results

Technique	Overall	Easy	Hard	Short	Medium	Long
GPT 4o Mini						
Baseline	29.3	31.1	28.2	31.8	28.6	26.2
Standard Pause Tokens	**30.2**	**34.6**	27.5	31.8	**28.7**	**30.4**
+ Instruction-Augmented Tokens	25.2	24.7	25.5	28.7	24.4	20.7
LLaMA 3.2 1B						
Baseline	22.9	20.3	24.4	26.7	**20.0**	22.2
Standard Pause Tokens	**23.1**	**21.4**	24.1	26.7	18.6	**25.9**
+ Instruction-Augmented Tokens	22.9	19.8	**24.8**	**27.2**	19.1	23.1
LLaMA 3.2 3B						
Baseline	26.4	25.0	27.3	27.8	23.3	30.6
Standard Pause Tokens	27.8	25.5	**29.3**	**28.3**	**24.7**	33.3
+ Instruction-Augmented Tokens	25.0	21.4	27.3	26.7	20.5	31.5
Pause-Tuning	**28.0**	**26.0**	**29.3**	28.9	24.2	**34.3**
LLaMA 3.1 8B						
Baseline	28.2	29.7	27.3	34.4	26.0	22.2
Standard Pause Tokens	29.2	32.8	27.0	**37.8**	25.1	23.1
+ Instruction-Augmented Tokens	29.6	32.8	**27.7**	36.1	26.0	**25.9**
Pause-Tuning	**29.8**	**34.9**	26.7	37.2	**27.0**	23.1
Phi 4 Mini						
Baseline	**28.2**	27.1	**28.9**	32.2	**23.3**	**31.5**
Standard Pause Tokens	28.0	**27.6**	28.3	31.1	**23.7**	**31.5**
+ Instruction-Augmented Tokens	27.2	**27.6**	27.0	**33.3**	22.8	25.9
Phi 4						
Baseline	27.4	27.6	27.3	31.1	23.7	28.7
Standard Pause Tokens	27.8	**28.6**	27.3	30.0	22.8	**34.3**
+ Instruction-Augmented Tokens	**29.0**	27.1	**30.2**	**33.3**	**24.2**	31.5
Gemma 3 12B						
Baseline	29.2	27.6	30.2	34.4	26.0	26.9
Standard Pause Tokens	**31.6**	**30.7**	**32.2**	**35.6**	**29.3**	**29.6**
+ Instruction-Augmented Tokens	30.2	27.6	31.8	**35.6**	28.8	24.1
GPT 4.1						
Baseline	46.7	52.1	43.4	**50.0**	47.4	39.8
Standard Pause Tokens	**47.9**	**54.2**	**44.1**	49.4	**49.3**	**42.6**
+ Instruction-Augmented Tokens	46.9	**54.2**	42.4	49.4	48.4	39.8
GPT 4.1 mini						
Baseline	29.6	30.2	29.3	37.8	26.0	23.1
Standard Pause Tokens	30.8	31.8	30.2	38.3	26.5	**26.9**
+ Instruction-Augmented Tokens	**31.6**	**33.3**	**30.5**	**40.0**	**29.8**	21.3
Claude 3.5 Sonnet						
Baseline	36.9	**51.2**	28.6	**46.7**	30.3	33.3
Standard Pause Tokens	**38.0**	42.9	**35.0**	43.4	**35.2**	**34.7**
+ Instruction-Augmented Tokens	34.7	38.2	32.5	45.6	29.2	26.8

Table 4: Evaluation results (%) of the LongBench v2 benchmark with various models and settings, comparing the baseline to Techniques 1, 2, and 5. Context Lengths are denoted as Short = 0-32K tokens, Medium = 32-64K tokens, Long = 64-128K tokens. The best result for each model in each seeting is highlighted in bold.

E. Needle in a Haystack Detailed Results

Technique	1K	2K	4K	8K	16K	32K	64K	128K
GPT 3.5								
Baseline	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.3 ± 0.6	-	-	-
Standard Pause Tokens	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.4 ± 0.2	-	-	-
+ Instruction-Augmented Tokens	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.4 ± 0.1	-	-	-
+ Pre-Prompt Instruction	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.4 ± 0.1	-	-	-
GPT 4o								
Baseline	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.8 ± 0.1	9.3 ± 0.2	8.8 ± 0.3
Standard Pause Tokens	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.7 ± 0.1	9.5 ± 0.0
+ Instruction-Augmented Tokens	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.5 ± 0.1
+ Pre-Prompt Instruction	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.7 ± 0.1	$\mathbf{9.6 \pm 0.1}$
LLaMA 3.2 1B								
Baseline	7.7 ± 0.9	6.6 ± 1.9	4.1 ± 1.9	3.7 ± 0.8	1.4 ± 0.4	1.4 ± 0.7	1.0 ± 0.1	$\mathbf{1.0 \pm 0.0}$
Standard Pause Tokens	7.8 ± 0.5	6.3 ± 0.3	4.1 ± 0.3	4.0 ± 1.4	2.0 ± 0.3	1.4 ± 0.9	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$
+ Instruction-Augmented Tokens	8.0 ± 0.7	6.6 ± 1.9	5.3 ± 2.9	3.1 ± 1.3	1.9 ± 0.8	1.0 ± 0.4	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$
+ Pre-Prompt Instruction	7.2 ± 0.6	7.6 ± 0.7	6.0 ± 2.3	3.7 ± 0.6	1.8 ± 0.3	1.7 ± 0.3	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$
LLaMA 3.2 3B								
Baseline	10.0 ± 0.0	9.9 ± 0.2	10.0 ± 0.0	9.1 ± 0.6	6.3 ± 1.4	3.8 ± 0.4	5.7 ± 1.8	5.2 ± 1.0
Standard Pause Tokens	10.0 ± 0.0	9.9 ± 0.1	9.7 ± 0.2	9.6 ± 0.3	6.5 ± 1.8	5.2 ± 0.5	5.6 ± 0.4	$\mathbf{5.6 \pm 0.5}$
+ Instruction-Augmented Tokens	10.0 ± 0.0	9.9 ± 0.1	9.9 ± 0.1	9.4 ± 0.4	6.4 ± 1.3	3.7 ± 0.4	5.4 ± 1.6	4.3 ± 1.5
+ Pre-Prompt Instruction	9.8 ± 0.4	9.9 ± 0.1	10.0 ± 0.0	9.6 ± 0.4	6.0 ± 2.2	5.8 ± 0.8	4.5 ± 1.3	3.3 ± 1.0
Pause-Tuning	10.0 ± 0.0	9.9 ± 0.1	9.9 ± 0.2	9.3 ± 0.5	7.9 ± 1.6	6.3 ± 1.7	5.9 ± 1.4	4.6 ± 0.4
LLaMA 3.1 8B								
Baseline	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.6 ± 0.7	9.4 ± 0.0	7.9 ± 1.4	7.4 ± 0.2
Standard Pause Tokens	10.0 ± 0.0	10.0 ± 0.0	9.7 ± 0.4	10.0 ± 0.0	9.6 ± 0.4	8.8 ± 0.9	8.4 ± 1.5	7.2 ± 0.7
+ Instruction-Augmented Tokens	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.8 ± 0.4	8.8 ± 0.6	7.5 ± 0.2	6.3 ± 0.9
+ Pre-Prompt Instruction	10.0 ± 0.0	10.0 ± 0.0	9.8 ± 0.4	9.8 ± 0.4	9.6 ± 0.4	9.2 ± 0.7	8.8 ± 0.5	7.2 ± 0.1
Pause-Tuning	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	9.4 ± 0.5	9.2 ± 0.3	$\mathbf{8.0 \pm 0.4}$

Table 5: Results of the single needle task, presented as mean $\pm$ standard deviation, comparing the baseline, pause token methods, and pause-tuning. Token counts are denoted as 1K = 1,000 tokens, 2K = 2,000 tokens, etc. The best result for 128K tokens is shown in bold.

F. Multi-Needle Results

While the results in the main paper utilize a single needle, we also conduct tests with three needles in the haystack. The results are reported in Appendix F.

Technique	1K	2K	4K	8K	16K	32K	64K	128K
GPT 3.5								
Baseline	10.0 ± 0.0	9.6 ± 1.1	9.9 ± 0.5	8.0 ± 1.7	8.0 ± 1.4	-	-	-
Standard Pause Tokens	10.0 ± 0.0	9.4 ± 1.2	9.9 ± 0.3	8.7 ± 2.2	7.7 ± 1.7	-	-	-
+ Instruction-Augmented Tokens	10.0 ± 0.0	10.0 ± 0.0	9.9 ± 0.5	8.5 ± 1.5	8.2 ± 1.4	-	-	-
+ Pre-Prompt Instruction	10.0 ± 0.0	9.6 ± 1.6	9.6 ± 1.1	8.9 ± 1.4	7.9 ± 2.1	-	-	-
GPT 4o								
Baseline	9.6 ± 1.1	10.0 ± 0.0	9.3 ± 1.6	8.8 ± 2.5	8.9 ± 2.3	7.2 ± 2.7	8.5 ± 2.6	7.6 ± 1.7
Standard Pause Tokens	9.2 ± 1.4	9.8 ± 0.8	9.6 ± 1.1	9.0 ± 1.5	8.5 ± 1.8	8.0 ± 2.5	7.8 ± 2.4	$\mathbf{8.4 \pm 1.9}$
+ Instruction-Augmented Tokens	10.0 ± 0.0	9.0 ± 1.1	8.6 ± 1.9	9.4 ± 1.2	8.0 ± 2.5	7.0 ± 3.2	7.7 ± 2.3	7.0 ± 2.3
+ Pre-Prompt Instruction	9.8 ± 0.8	10.0 ± 0.0	9.2 ± 1.8	9.4 ± 1.2	8.8 ± 1.5	8.2 ± 1.9	7.0 ± 2.3	7.2 ± 1.4
LLaMA 3.2 1B								
Baseline	6.4 ± 4.6	8.6 ± 3.2	7.0 ± 4.4	1.0 ± 0.0	3.6 ± 4.1	1.0 ± 0.0	2.2 ± 3.2	$\mathbf{1.0 \pm 0.0}$
Standard Pause Tokens	8.2 ± 3.7	9.4 ± 2.3	9.4 ± 2.3	6.2 ± 4.5	5.0 ± 4.5	3.4 ± 4.1	2.8 ± 3.7	$\mathbf{1.0 \pm 0.0}$
+ Instruction-Augmented Tokens	3.0 ± 1.9	2.2 ± 1.9	2.8 ± 2.7	2.2 ± 2.2	1.4 ± 1.1	1.8 ± 2.4	1.6 ± 1.7	$\mathbf{1.0 \pm 0.0}$
+ Pre-Prompt Instruction	2.4 ± 1.6	2.2 ± 1.5	2.4 ± 1.6	2.8 ± 1.9	1.4 ± 1.1	1.6 ± 1.7	1.4 ± 1.1	$\mathbf{1.0 \pm 0.0}$
LLaMA 3.2 3B								
Baseline	7.2 ± 3.1	6.2 ± 2.4	5.2 ± 2.2	3.6 ± 1.9	4.2 ± 2.1	3.4 ± 1.7	3.4 ± 2.3	2.2 ± 1.5
Standard Pause Tokens	8.0 ± 2.2	7.4 ± 1.9	6.1 ± 2.4	5.5 ± 1.9	4.8 ± 1.4	4.9 ± 1.8	4.3 ± 0.8	$\mathbf{3.1 \pm 2.7}$
+ Instruction-Augmented Tokens	7.1 ± 2.6	5.7 ± 2.7	5.3 ± 1.3	4.3 ± 1.4	5.0 ± 2.1	4.1 ± 1.2	3.7 ± 1.1	3.0 ± 2.5
+ Pre-Prompt Instruction	7.2 ± 2.7	7.0 ± 2.3	6.0 ± 1.5	4.4 ± 1.9	4.6 ± 2.3	4.0 ± 1.6	4.2 ± 1.4	3.0 ± 1.5
Pause-Tuning	7.0 ± 3.6	5.8 ± 2.7	4.8 ± 4.0	2.8 ± 3.0	1.2 ± 4.9	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
LLaMA 3.1 8B								
Baseline	8.1 ± 2.7	7.5 ± 1.9	6.8 ± 1.8	6.4 ± 1.7	4.4 ± 2.1	4.3 ± 1.6	4.3 ± 0.8	4.2 ± 0.8
Standard Pause Tokens	8.0 ± 2.7	7.4 ± 1.6	6.7 ± 2.5	7.0 ± 2.0	5.2 ± 1.5	5.0 ± 1.5	4.5 ± 1.6	4.2 ± 1.4
+ Instruction-Augmented Tokens	9.0 ± 1.5	7.7 ± 1.5	7.5 ± 1.4	7.1 ± 1.0	6.7 ± 0.7	6.9 ± 0.5	6.6 ± 1.6	$\mathbf{5.7 \pm 1.7}$
+ Pre-Prompt Instruction	8.2 ± 1.5	7.4 ± 2.0	7.6 ± 1.2	7.1 ± 1.2	7.3 ± 0.8	7.2 ± 0.8	7.0 ± 0.0	5.3 ± 2.3
Pause-Tuning	8.3 ± 2.3	7.9 ± 2.7	8.0 ± 2.5	7.1 ± 2.5	6.0 ± 2.1	5.8 ± 2.3	4.7 ± 2.6	4.3 ± 2.7

G. Longbench Domain Results

Model	Overall	Easy	Hard	Short	Medium	Long
Single-Document QA						
GPT-4.1 Baseline	44.6	50.0	41.4	50.0	41.6	37.5
GPT-4.1 (Std. Pause)	**46.3**	**54.7**	41.4	50.0	**45.5**	37.5
GPT-4.1 (+ Instr.)	**46.3**	50.0	**44.1**	**51.4**	42.9	**41.7**
GPT-4.1-mini Baseline	32.6	31.2	33.3	45.9	20.8	29.2
GPT-4.1-mini (Std. Pause)	34.3	32.8	35.1	48.6	23.4	25.0
GPT-4.1-mini (+ Instr.)	35.4	35.9	35.1	**51.4**	26.0	16.7
Multi-Document QA						
GPT-4.1 Baseline	**47.2**	52.9	**45.1**	**51.7**	**50.0**	**30.4**
GPT-4.1 (Std. Pause)	45.6	**55.9**	41.8	**51.7**	47.7	26.1
GPT-4.1 (+ Instr.)	42.4	**55.9**	37.4	46.6	45.5	26.1
GPT-4.1-mini Baseline	32.8	44.1	28.6	39.7	27.3	26.1
GPT-4.1-mini (Std. Pause)	30.4	41.2	26.4	39.7	22.7	21.7
GPT-4.1-mini (+ Instr.)	30.4	41.2	26.4	36.2	29.5	17.4
Long In-context Learning						
GPT-4.1 Baseline	56.8	**63.9**	**55.6**	**58.3**	**58.1**	**57.7**
GPT-4.1 (Std. Pause)	56.8	58.3	**55.6**	50.0	**58.1**	**57.7**
GPT-4.1 (+ Instr.)	**58.0**	61.1	**55.6**	**58.3**	**58.1**	**57.7**
GPT-4.1-mini Baseline	35.8	33.3	37.8	41.7	37.2	30.8
GPT-4.1-mini (Std. Pause)	34.6	30.6	37.8	33.3	**39.5**	26.9
GPT-4.1-mini (+ Instr.)	30.9	30.6	31.1	33.3	37.2	19.2
Code Repository Understanding						
GPT-4.1 Baseline	48.0	**55.6**	43.8	**58.3**	**55.6**	41.4
GPT-4.1 (Std. Pause)	**54.0**	**55.6**	**53.1**	**58.3**	44.4	**55.2**
GPT-4.1 (+ Instr.)	46.0	**55.6**	40.6	**58.3**	**55.6**	37.9
GPT-4.1-mini Baseline	28.0	27.8	28.1	**58.3**	33.3	20.7
GPT-4.1-mini (Std. Pause)	36.0	27.8	40.6	41.7	33.3	34.5
GPT-4.1-mini (+ Instr.)	38.0	33.3	40.6	**58.3**	33.3	31.0
Long-dialogue History Understanding						
GPT-4.1 Baseline	**41.0**	**50.0**	**31.6**	40.0	**42.1**	**0**
GPT-4.1 (Std. Pause)	**41.0**	**50.0**	**31.6**	40.0	**42.1**	**0**
GPT-4.1 (+ Instr.)	**41.0**	**50.0**	**31.6**	**45.0**	36.8	**0**
GPT-4.1-mini Baseline	15.4	25.0	5.3	0.0	31.6	**0**
GPT-4.1-mini (Std. Pause)	15.4	20.0	10.5	5.0	26.3	**0**
GPT-4.1-mini (+ Instr.)	25.6	30.0	21.1	10.0	**42.1**	**0**
Long Structured Data Understanding						
GPT-4.1 Baseline	36.4	35.0	**38.5**	**25.0**	47.8	0.0
GPT-4.1 (Std. Pause)	42.4	45.0	**38.5**	**25.0**	56.5	0.0
GPT-4.1 (+ Instr.)	**48.5**	**55.0**	**38.5**	**25.0**	**60.9**	16.7
GPT-4.1-mini Baseline	12.1	15.0	7.7	**25.0**	13.0	0.0
GPT-4.1-mini (Std. Pause)	15.2	20.0	7.7	0.0	13.0	**33.3**
GPT-4.1-mini (+ Instr.)	15.2	20.0	7.7	0.0	17.4	16.7

Table 6: Domain-specific performance (%) on LongBench v2 for GPT-4.1 and GPT-4.1-mini models. This table breaks down results across six distinct domains, comparing baseline models with their pause-tuned variants. The best result in each column for each domain is highlighted in **bold**.

Model	Overall	Easy	Hard	Short	Medium	Long
Single-Document QA						
Claude 3.5 Sonnet Baseline	38.0	**58.1**	26.9	**48.1**	29.1	35.7
Claude 3.5 Sonnet (Std. Pause)	**40.1**	53.4	**32.3**	44.9	**32.4**	**50.0**
Claude 3.5 Sonnet (+ Instr.)	34.4	43.8	28.4	44.2	25.9	31.2
Multi-Document QA						
Claude 3.5 Sonnet Baseline	43.8	**65.0**	36.7	**50.0**	43.3	28.6
Claude 3.5 Sonnet (Std. Pause)	**46.0**	54.5	**42.5**	45.3	**48.7**	**42.9**
Claude 3.5 Sonnet (+ Instr.)	40.2	43.5	39.1	**50.0**	40.0	17.6
Long In-context Learning						
Claude 3.5 Sonnet Baseline	**38.3**	**46.2**	32.4	50.0	34.5	**38.1**
Claude 3.5 Sonnet (Std. Pause)	38.0	37.1	38.6	50.0	**40.5**	28.0
Claude 3.5 Sonnet (+ Instr.)	37.0	34.8	**39.1**	**55.6**	36.4	26.7
Code Repository Understanding						
Claude 3.5 Sonnet Baseline	**44.1**	**69.2**	28.6	**77.8**	20.0	**35.0**
Claude 3.5 Sonnet (Std. Pause)	35.6	46.7	**30.0**	55.6	**33.3**	29.6
Claude 3.5 Sonnet (+ Instr.)	37.1	41.7	34.8	71.4	**33.3**	30.0
Long-dialogue History Understanding						
Claude 3.5 Sonnet Baseline	16.7	**23.1**	9.1	14.3	20.0	**0**
Claude 3.5 Sonnet (Std. Pause)	**26.3**	20.0	**33.3**	26.3	**26.3**	**0**
Claude 3.5 Sonnet (+ Instr.)	24.1	20.0	28.6	**28.6**	20.0	**0**

Table 7: Domain-specific performance (%) on LongBench v2 for Claude 3.5 Sonnet model. This table breaks down results across various distinct domains, comparing baseline models with their pause-tuned variants. The best result in each column for each domain is highlighted in **bold**.