Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Why LLMs Refuse to Summarize More in Long Contexts?

Musa Cim

mtc5693@psu.edu

Pennsylvania State University

Burak Topcu

Pennsylvania State University

Mahmut Kandemir

Pennsylvania State University

Research Article

Keywords: large language models, summarization, context compaction, long context, LLM agents

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11639533/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Why LLMs Refuse to Summarize More in Long Contexts?

Musa Cim, Burak Topcu, Mahmut Taylan Kandemir
The Pennsylvania State University
University Park, PA, USA
{mtc5693, bvt5283, mtk2}@psu.edu

***Abstract*—Long-horizon agents routinely exceed their context windows. A common form of context compaction that reduces a long interaction history to a smaller representation is to summarize the history and continue from the summary. This summarization loses information, and this loss is a common source of agent failures: forgotten details are rediscovered at extra cost or silently dropped. The natural repair is to ask for a longer summary, so we investigate whether that option actually works. Does summary length grow with input length? Across eleven models spanning coder, general-purpose, and reasoning families, summary length grows far more slowly than the input, so at 100k tokens, less than one percent of the input survives on both long documents and real agent trajectories. Furthermore, prompt-level directives do not close this gap: explicit length directives raise output only modestly, and the summary length plateaus far below the requested amount, however the request is phrased. To test whether training data can move this ceiling, we lightly fine-tune two models on summaries that differ only in length. The resulting output length shifts dramatically in either direction, tracking the fine-tuning summary length, even under a prompt that never mentions length. Consequently, this paper shows that summary length grows strongly sublinearly with input size, is only weakly responsive to length directives in the prompt, and is strongly responsive to fine-tuning, suggesting that training data, rather than the prompt, is the effective lever.**

Code and data:
https://github.com/musacim/long-context-summarization

***Index Terms*—large language models, summarization, context compaction, long context, LLM agents**

I. Introduction

Autonomous agents are increasingly deployed on long-horizon workloads [1] such as software engineering [2], [3], research [4], and computer use [5], where every action, observation, and tool result accumulates in the conversation history [6]–[8]. Current models offer context windows of roughly 128k to 1M tokens [9], and long-running tasks routinely outgrow them. Larger windows would not solve the problem either, since models lose track of information in the middle of long inputs and perform worse as inputs grow [10], [11]. Working beyond the context window therefore requires context management. An agent remembers only what its context still holds. Deployed agent frameworks such as Claude Code [12], OpenHands [13], and OpenAI Codex [14] answer this with *context compaction*: when the number of tokens accumulated in the context approaches the context-window limit, the model summarizes it, and inference continues from the summary [15]. Because compaction runs through ordinary model calls at the application layer, it is now the default context-management operator in several major products. Its efficiency lies in both the low cost of one extra summarization call and the brevity of the resulting summary, which is useful as long as it adequately substitutes for what it replaced [16]. Summary length is therefore the critical quantity, as it determines compaction quality and how much of the replaced content survives.

The length of a compaction summary directly affects the quality of the retained context, with its impact becoming more pronounced in long-horizon tasks [17]. For short contexts, information lost during a single compaction may have a limited impact on overall service quality. However, as tasks grow longer and require repeated compactions, these losses can accumulate and propagate: information omitted by one summary, such as a file path, a failed attempt, or a constraint the user stated earlier, is unavailable to subsequent steps and may be further lost in later compactions. An agent that loses such information must either spend extra calls rediscovering it or proceed without it [18]. If summary length grew with the history, or at least responded to a request for more detail, this loss would be a cost that frameworks could tune. Because there is no reliable way to control summary length in advance [19], frameworks that rely on compaction are effectively constrained by the model's default summarization behavior. Since summary length bounds the information a summary can retain without requiring a reference or judge, which can be unreliable for long inputs [16], we measure summary length as a necessary condition for high-quality compaction.

We study summarization length over long-horizon contexts in three controlled steps. First, we measure how summary length responds to input length across nested prefixes of long documents and real agent trajectories, ensuring that only input length changes. Second, we hold the input fixed at 100k tokens and vary only the prompt-level length directive. Third, we hold the prompt fixed and vary only the length of the summaries on which a model is fine-tuned.

This paper reports three findings.

- **Summary length grows strongly sublinearly with input length.** Across eleven models, summaries of long-document QA inputs and agentic trajectories remain within a narrow band even as input grows $100\times$.
- **Prompt-level length directives do not lift the ceiling.** Additional prompt directives increase output length beyond

TABLE I
SUMMARY LENGTH VS. INPUT LENGTH ACROSS ELEVEN MODELS.

	Model	Domain	Input length (tokens) 1k	2k	4k	8k	16k	32k	64k	100k
Coder models	Qwen2.5-Coder-7B [20]	QA	184 ± 89	239 ± 141	311 ± 206	289 ± 139	315 ± 173	443 ± 255	428 ± 383	484 ± 465
		Agentic	94 ± 72	318 ± 79	482 ± 130	527 ± 168	597 ± 194	554 ± 184	674 ± 326	692 ± 379
	DeepSeek-Coder-V2-Lite [21]	QA	220 ± 105	218 ± 99	176 ± 99	281 ± 139	303 ± 108	498 ± 324	366 ± 307	207 ± 131
		Agentic	419 ± 392	315 ± 93	332 ± 111	401 ± 130	446 ± 121	825 ± 372	399 ± 226	237 ± 105
General-purpose non-reasoning	Llama-3.1-8B [22]	QA	262 ± 53	301 ± 50	361 ± 69	417 ± 92	500 ± 247	590 ± 292	747 ± 457	1181 ± 977
		Agentic	201 ± 83	322 ± 89	462 ± 168	508 ± 156	563 ± 182	688 ± 247	683 ± 252	526 ± 201
	Llama-3.3-70B [22]	QA	232 ± 56	280 ± 33	344 ± 59	394 ± 56	454 ± 77	452 ± 119	512 ± 135	749 ± 578
		Agentic	111 ± 15	274 ± 76	385 ± 89	363 ± 83	366 ± 88	346 ± 94	376 ± 126	512 ± 260
	Qwen2.5-7B [20]	QA	271 ± 114	253 ± 114	313 ± 132	400 ± 205	539 ± 163	602 ± 242	632 ± 371	975 ± 956
		Agentic	219 ± 66	255 ± 106	649 ± 381	1058 ± 809	663 ± 150	624 ± 163	646 ± 225	846 ± 408
	Qwen3-30B-A3B-Instruct [23]	QA	416 ± 197	548 ± 182	653 ± 178	833 ± 189	1045 ± 294	1126 ± 342	1157 ± 291	1156 ± 344
		Agentic	80 ± 12	413 ± 99	843 ± 898	511 ± 155	578 ± 184	542 ± 114	609 ± 150	635 ± 256
	Phi-3.5-MoE [24]	QA	191 ± 63	198 ± 44	220 ± 72	234 ± 73	268 ± 84	285 ± 115	303 ± 150	264 ± 137
		Agentic	251 ± 90	399 ± 517	391 ± 96	391 ± 83	386 ± 87	472 ± 140	472 ± 130	438 ± 114
General-purpose reasoning	gpt-oss-20b [25]	QA	317 ± 142	406 ± 195	575 ± 157	684 ± 230	764 ± 238	719 ± 231	865 ± 326	969 ± 333
		Agentic	27 ± 13	216 ± 109	238 ± 118	251 ± 127	256 ± 186	315 ± 283	333 ± 363	230 ± 239
	gpt-oss-120b [25]	QA	494 ± 211	668 ± 255	664 ± 229	920 ± 280	1089 ± 310	1267 ± 406	1316 ± 579	1233 ± 500
		Agentic	65 ± 25	324 ± 90	491 ± 145	519 ± 185	714 ± 275	639 ± 262	1039 ± 322	989 ± 339
	Qwen3-30B-A3B-Thinking [23]	QA	416 ± 71	493 ± 122	583 ± 116	664 ± 135	739 ± 129	664 ± 237	631 ± 310	685 ± 306
		Agentic	47 ± 24	210 ± 79	280 ± 99	351 ± 98	325 ± 88	368 ± 113	437 ± 98	451 ± 98
	DeepSeek-R1-Distill-Llama-8B [26]	QA	192 ± 77	210 ± 100	234 ± 92	292 ± 158	336 ± 123	408 ± 163	496 ± 165	339 ± 208
		Agentic	199 ± 144	466 ± 660	684 ± 577	872 ± 967	818 ± 624	851 ± 651	701 ± 471	660 ± 502
Mean output/input ratio			22.3%	16.7%	11.0%	6.34%	3.43%	**1.89%**	**0.98%**	**0.66%**

the uninstructed baseline, but it plateaus at a few thousand tokens, irrespective of whether the target is specified in tokens, words, or characters.

- **Fine-tuning moves the ceiling in both directions.** Light LoRA training on summaries that differ only in length makes the same model write far shorter or far longer summaries under a neutral prompt, "Summarize the following document."

Together, these results point away from the input and the prompt directives and toward training: summary length behaves more like a learned default than a quantity planned from the input. For agent frameworks, asking for a longer summary is thus not a reliable way to reduce compaction loss, and the fix lies in the data used to train the summarizer.

II. SUMMARY LENGTH GROWS FAR MORE SLOWLY THAN INPUT

a) Models and benchmarks: We evaluate the eleven open-weight models listed in Table I, spanning coder, general-purpose, and reasoning families, both dense and mixture-of-experts, from 7B to 120B parameters. Inputs come from two sources: i) long documents from LongBench-v2 [27], and ii) real OpenHands coding-agent trajectories from Open-SWE-Traces [28]. LongBench-v2 contains long, largely self-contained contexts drawn from diverse sources, including books, reports, papers, dialogues, code repositories, and structured data. Agent trajectories are what compaction sees in deployment, with OpenHands providing representative examples that interleave the model's reasoning, tool calls such as shell commands and file edits, and tool output such as file contents, logs, and error messages. Much of it is raw tool output repeated across steps, while what matters later, namely what was tried, what failed, and what changed, is sparse, so a good summary must track state, not retell content.

b) Experimental setup: From each source, we take ten documents and build *nested* chains at eight input lengths from 1k to 100k tokens, where each shorter input is a literal prefix of the longer ones. Using nested prefixes is important because it avoids confounding input length with document identity or content. Moving from 8k to 16k tokens, for example, only appends additional context to the exact same underlying input. For every model, document, and input length, we collect three sampled summaries, and Table I reports their mean and standard deviation. The summarization prompt we used for our QA experiments is neutral: "Summarize the following document."; the agentic prompt is the production OpenHands LLMSUMMARIZINGCONDENSER prompt [13] verbatim. Neither states a target length. The last row of Table I reports the mean output-to-input ratio over all models on both QA and agentic inputs. We exclude the 0.7% of all runs that looped until they hit the generation limit.

c) Results: As the input grows from 1k to 100k tokens, mean summary length grows far more slowly than the input: every model stays below roughly $1{,}250$ tokens even at 100k,

TABLE II
OUTPUT LENGTH UNDER EXPLICIT LENGTH INSTRUCTIONS AT 100K INPUT.

Instruction	N	Target	Qwen2.5-7B		Llama-3.1-8B		gpt-oss-20b	
			Output	Achieved	Output	Achieved	Output	Achieved
No prompt-level directive for length	—	—	827	—	782	—	534	—
"at least N tokens"	2,000	2,000	1,718	86%	1,078	54%	2,453	**123%**
	4,000	4,000	1,916	48%	1,359	34%	2,915	73%
	8,000	8,000	2,625	33%	1,288	16%	2,246	28%
	16,000	16,000	2,364	15%	872	5%	1,060	7%
"at least N words"	1,250	2,000	2,177	**115%**	1,106	68%	1,928	**106%**
	2,500	4,000	2,367	61%	1,128	36%	3,459	92%
	5,000	8,000	2,802	35%	1,115	17%	3,470	43%
	9,950	16,000	3,024	17%	1,042	8%	2,383	15%
"at least N characters"	8,900	2,000	2,110	**105%**	1,018	54%	2,585	**125%**
	17,800	4,000	2,163	54%	1,321	35%	2,393	60%
	35,600	8,000	1,581	19%	1,140	15%	1,875	23%
	71,200	16,000	1,754	11%	1,209	8%	1,719	11%
"at least N as long"	one quarter	25,000	798	3%	764	3%	1,661	7%
	one third	33,333	943	3%	692	2%	1,444	4%
	half	50,000	799	2%	872	2%	1,343	3%

as given in Table I. The mean output-to-input ratio falls from 22.3% at 1k to 0.66% at 100k, so at 100k more than 99% of the input is no longer retained. Long documents and agent trajectories do contain repetition, and thus, many parts of the context can be summarized safely. However, when the resulting summary contains less than one percent of the original input, roughly one sentence survives for every hundred, leaving little room to preserve the details needed for subsequent steps. The same pattern holds across model families, whether instruct or coder, reasoning or non-reasoning.

On agentic inputs, growth stops early: eight of the eleven models write at most 30% more at 100k than at 8k. Length is not even monotonic: for about half the models, the 100k summary is shorter than the 32k or 64k one, so more input does not buy more output. Model scale does not help either, as Llama-3.3-70B writes shorter summaries than Llama-3.1-8B, and post-training shifts only the level: Qwen3-30B-A3B writes 1,156 tokens at 100k as an instruct model and 685 as a thinking model, and both flatten. Long inputs also make length less predictable: at 100k, several models have standard deviations near the mean, as for Qwen2.5-7B on QA (975 ± 956), which complicates any fixed compaction budget.

Takeaway #1: Across model types (i.e., coder, instruct, reasoning, and non-reasoning models) and scales, summary length grows strongly sublinearly with input length.

III. ASKING FOR MORE DOES NOT PRODUCE MORE

a) Experimental setup: The prompts in Section II contain no prompt-level directives for summary length, so the flat summary lengths there might simply mean that models need an explicit one. We therefore evaluate explicit summary-length directives here. We use only the 100k-token inputs and vary the length directive alone, testing Qwen2.5-7B, Llama-3.1-8B, and gpt-oss-20b, three different model families chosen for coverage, on three QA documents and three agentic trajectories, with three samples each. Each prompt-directive is written three ways: in tokens, in words, and in characters, since the choice of unit may itself affect the response. The word and character versions of each request ask for the same amount of text as the token version, using conversion rates measured over model-generated summaries, roughly 0.6 words or 4.5 characters per token, consistent with common tokenizer rules of thumb [29]. We additionally include *relative* requests for a quarter, a third, and half of the document.

In Table II, *Target* is the requested amount expressed in tokens, *Output* corresponds to the mean number of tokens the model delivered, and *Achieved* is the delivered amount measured in the requested unit, as a percentage of the request. We exclude the 0.9% of all runs that looped until they hit the generation limit. All remaining outputs end well before the limit, so the models stopped on their own.

b) Results: As Table II shows, explicit prompt-level directives raise output above the no-directive baseline, but the delivered amount still does not scale with the requested amount. Qwen2.5-7B delivers between roughly 1,600 and 3,000 tokens for every explicit request from 2,000 to 16,000 tokens; raising the request leaves the delivered amount unchanged, so the Achieved column falls from 86% to 15% only because the request itself grows. Llama-3.1-8B plateaus lower, never exceeding about 1,400 tokens at any request, and gpt-oss-20b peaks near 3,500. The three units produce outputs of similar magnitude at each requested amount. Furthermore, *relative* summary length requests for a quarter, a third, or half of the document are the least effective of all. Under these relative

requests, Qwen2.5-7B and Llama-3.1-8B fall back to a directive-free baseline, and gpt-oss-20b, though it stays above its baseline, still delivers at most 7% of the request. Since the inputs are 100k tokens, even the best instructed outputs amount to roughly three percent of the input. Without summary length directives, the ratio in Section II was 0.66%, so directives provide a slight increase, but the information loss remains at a significant level.

c) Which unit works best?: Words work best for Qwen2.5-7B and gpt-oss-20b, although no unit reaches the larger requests. Averaged over the four requested amounts, word requests yield 2,592 tokens for Qwen2.5-7B and 2,810 for gpt-oss-20b, against 2,156 and 2,168 for token requests and 1,902 and 2,143 for character requests. Words are also the only unit for which Qwen2.5-7B's output keeps rising with the request, from 2,177 to 3,024 tokens, and at the largest request, they give gpt-oss-20b more than twice the output of token requests (2,383 versus 1,060). Characters perform worst at large summary targets: both models deliver less for 35,600 and 71,200 characters than for 8,900. Llama-3.1-8B is insensitive to the unit, averaging between 1,098 and 1,172 tokens under all three. We hypothesize that words are the most common unit of length in directives given to the model during training, whereas token counts rarely appear in human-written directives, and character counts are hard to estimate. Standard decoding further gives the model no running count of the tokens it has produced; adding an explicit countdown makes length control far more precise [30].

Takeaway #2: Prompt-level length directives are only partially followed: output increases marginally and remains far below the requested amount.

TABLE III
SUMMARY LENGTH (OUTPUT TOKENS) BEFORE AND AFTER FINE-TUNING. PARENTHESES GIVE THE LENGTH OF THE FINE-TUNING SUMMARIES.

		Qwen2.5-7B		Llama-3.1-8B	
	Supervision	median	(shortest–longest)	median	(shortest–longest)
	Base model	690	(308–1,306)	605	(465–869)
Fine-tuned	short (134 tok)	**102**↓	(80–177)	**133**↓	(67–274)
	medium (446 tok)	327↓	(219–743)	340↓	(195–633)
	long (1,999 tok)	**1,284**↑	(695–2,574)	**1,139**↑	(576–1,907)

IV. FINE-TUNING MOVES THE CEILING IN BOTH DIRECTIONS

a) Experimental setup: In previous sections, we explored the impact of context length itself and prompt-level summary directives on summary length. Here, we further investigate whether training data limits the achievable summary length. We build three fine-tuning sets over the same 100 source documents, using 32k-token prefixes of distinct books from InfiniteBench [31]. Each set is paired with short, medium, or long reference summaries of median length 134, 446, and 1,999 tokens, respectively, generated with Claude Sonnet 5. Thus, the target length is the only difference between the sets.

We train LoRA adapters [32] for two epochs on each set for Qwen2.5-7B and Llama-3.1-8B, with **the neutral prompt** and **no length directive**, and evaluate on held-out LongBench-v2 documents using greedy decoding. Table III reports the smallest, median, and longest generated lengths for the base model and each set.

b) Results: Fine-tuning moves the summary length in both directions. Short supervision drives the Qwen2.5-7B median from 690 tokens to 102, long supervision raises it to 1,284, and Llama-3.1-8B follows the same pattern, from 605 to 133 and 1,139. Both shifts are statistically significant for both models. A neutral summary prompt that produced comparably small summary lengths for much larger context lengths in Section II now produces far shorter or far longer summaries depending only on the training examples.

Takeaway #3: Training-data summary length shapes output length: models fine-tuned on shorter or longer summaries produce correspondingly shorter or longer outputs, even without an explicit length directive in the prompt.

V. RELATED WORK

Context management for agents extends beyond summarization. Proposed methods span several families: heuristic truncation and sliding windows [33], prompt compression [34], KV cache compression [35], memory systems [36], and learned history compression for agents [37]. Several of these operate on model internals or inside the serving stack, so they are not usable by agent frameworks that run on hosted model APIs, and deployed frameworks commonly fall back to simple summarization.

A second line of work studies making models write longer on open-ended prompts such as essay writing, where dedicated fine-tuning sets can extend outputs to many thousands of words [38], and benchmarks report that length instructions are followed worse as inputs grow [39]. In contrast, we study summarization, where the model condenses a long given input rather than writing freely.

Another approach to generating summaries of long inputs is divide and conquer, which splits the input into chunks and summarizes them separately or recursively [40]–[42]. This approach does not suit autonomous agents, for two main reasons. First, chunks are summarized independently, which breaks the dependencies between steps that an agent needs to interpret later actions. Second, chunking breaks the *prefix cache*. Agentic traces accumulate, and serving systems reuse the cached prefix as a major cost saver, so every chunk call presents a fresh prefix that must be prefilled and paid for from scratch.

VI. Discussion and Future Work

a) Why agentic summaries are different: Compaction must decide what to keep before knowing what the agent will need next. Summarization training data teaches a different habit: a good document summary retells the content, regroups it by theme, and drops detail in favor of the main line. An agent trajectory is instead a chronological log in which any step may matter later, so its summary should preserve the order of events, treat steps evenly rather than reframing them around a theme, and keep exact details such as file paths, commands, and error messages. A summarizer trained to retell documents is therefore mismatched to compaction, which may explain why trajectories received the shortest summaries in our experiments.

b) Limitations and future work: We measure summary length, not summary content. Length is necessary for retaining detail but not sufficient, so a natural next step is to measure information retention and downstream agent success as a function of summary length. Our experiments cover open-weight models only; whether hosted models used in production frameworks behave the same way remains to be tested. We also run a single compaction round, whereas a long-running agent compacts repeatedly, and whatever one summary omits is unavailable to every later step and cannot be recovered by later rounds. Losses therefore likely compound across rounds, and measuring this directly is left for future work. Finally, our fine-tuning uses fixed 32k-token inputs and fixed-length targets, so it does not test whether a model can learn to scale summary length with input length. Training on summaries whose length grows with the input is a natural next step.

VII. Conclusion

We studied the most common context compaction method, summarization by the model itself, and analyzed how summary length behaves. Across eleven models, input length had little effect on the length of the generated summary, explicit summary length directives were only partially followed, and the length of the training summaries had a visible impact. We hope these findings inform the design of compaction mechanisms in agent frameworks and the training of models for long-horizon context management.

VIII. Acknowledgment

This research used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357.

References

[1] T. Kwa, B. West, J. Becker, A. Deng, K. Garcia, M. Hasin, S. Jawhar, M. Kinniment, N. Rush, S. von Arx *et al.*, "Measuring AI ability to complete long software tasks," 2025. [Online]. Available: http://papers.nips.cc/paper_files/paper/2025/hash/85069585133c4c168c865e65d72e9775-Abstract-Conference.html

[2] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. R. Narasimhan, "Swe-bench: Can language models resolve real-world github issues?" in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=VTF8yNQM66

[3] X. Deng, J. Da, E. Pan, Y. Y. He, C. Ide, K. Garg, N. Lauffer, A. Park, N. Pasari, C. Rane *et al.*, "Swe-bench pro: Can AI agents solve long-horizon software engineering tasks?" *CoRR*, vol. abs/2509.16941, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.16941

[4] J. Wei, Z. Sun, S. Papay, S. McKinney, J. Han, I. Fulford, H. W. Chung, A. T. Passos, W. Fedus, and A. Glaese, "Browsecomp: A simple yet challenging benchmark for browsing agents," *CoRR*, vol. abs/2504.12516, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.12516

[5] T. Xie, D. Zhang, J. Chen, X. Li, S. Zhao, R. Cao, T. J. Hua, Z. Cheng, D. Shin, F. Lei *et al.*, "Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments," 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html

[6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," 2023. [Online]. Available: https://openreview.net/forum?id=WE_vluYUL-X

[7] J. Yang, C. E. Jimenez, A. Wettig, K. Lieret, S. Yao, K. Narasimhan, and O. Press, "Swe-agent: Agent-computer interfaces enable automated software engineering," 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html

[8] X. Wang, B. Li, Y. Song, F. F. Xu, X. Tang, M. Zhuge, J. Pan, Y. Song, B. Li, J. Singh *et al.*, "Openhands: An open platform for AI software developers as generalist agents," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=OJd3ayDDoF

[9] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. P. Lillicrap, J. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *CoRR*, vol. abs/2403.05530, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.05530

[10] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 157–173, 2024. [Online]. Available: https://doi.org/10.1162/tacl_a_00638

[11] C. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, Y. Zhang, and B. Ginsburg, "RULER: what's the real context size of your long-context language models?" *CoRR*, vol. abs/2404.06654, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2404.06654

[12] Anthropic, "Context compaction," https://platform.claude.com/docs/en/build-with-claude/compaction, 2026.

[13] OpenHands, "LLMSummarizingCondenser," https://docs.openhands.dev/sdk/arch/condenser, 2026.

[14] OpenAI, "Context compaction," https://developers.openai.com/api/docs/guides/compaction, 2026.

[15] T. Lindenbauer, I. Slinko, L. Felder, E. Bogomolov, and Y. Zharov, "The complexity trap: Simple observation masking is as efficient as LLM summarization for agent context management," *CoRR*, vol. abs/2508.21433, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.21433

[16] Y. Kim, Y. Chang, M. Karpinska, A. Garimella, V. Manjunatha, K. Lo, T. Goyal, and M. Iyyer, "FABLES: evaluating faithfulness and content selection in book-length summarization," *CoRR*, vol. abs/2404.01261, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2404.01261

[17] Y. Li, Z. Hou, Y. Jing, J. Tang, and Y. Dong, "Compactionrl: Reinforcement learning with context compaction for long-horizon agents," 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2607.05378

[18] M. Cemri, M. Z. Pan, S. Yang, L. A. Agrawal, B. Chopra, R. Tiwari, K. Keutzer, A. G. Parameswaran, D. Klein, K. Ramchandran *et al.*, "Why do multi-agent LLM systems fail?" 2025. [Online]. Available: http://papers.nips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html

[19] F. Retkowski and A. Waibel, "Zero-shot strategies for length-controllable summarization," in *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, ser. Findings of ACL, L. Chiruzzo, A. Ritter, and L. Wang, Eds., vol. NAACL 2025. Association for Computational Linguistics, 2025, pp. 551–572. [Online]. Available: https://doi.org/10.18653/v1/2025.findings-naacl.34

[20] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Dang *et al.*, "Qwen2.5-coder technical

report," *CoRR*, vol. abs/2409.12186, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2409.12186

[21] DeepSeek-AI, Q. Zhu, D. Guo, Z. Shao, D. Yang, P. Wang, R. Xu, Y. Wu, Y. Li, H. Gao *et al.*, "Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence," *CoRR*, vol. abs/2406.11931, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.11931

[22] L. Team, "The llama 3 herd of models," *CoRR*, vol. abs/2407.21783, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.21783

[23] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

[24] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," 2024. [Online]. Available: https://arxiv.org/abs/2404.14219

[25] OpenAI, "gpt-oss-120b & gpt-oss-20b model card," https://openai.com/index/gpt-oss-model-card/, 2025, accessed: 2026-09-28.

[26] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi *et al.*, "Deepseek-r1 incentivizes reasoning in llms through reinforcement learning," *Nature*, vol. 645, no. 8081, p. 633–638, 2025. [Online]. Available: http://dx.doi.org/10.1038/s41586-025-09422-z

[27] Y. Bai, S. Tu, J. Zhang, H. Peng, X. Wang, X. Lv, S. Cao, J. Xu, L. Hou, Y. Dong *et al.*, "Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 3639–3664. [Online]. Available: https://doi.org/10.18653/v1/2025.acl-long.183

[28] W. U. Ahmad, N. Ludwig, S. Majumdar, and B. Ginsburg, "Open-swe-traces: Advancing dual-mode multilingual distillation for software engineering agents," *CoRR*, vol. abs/2606.16038, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2606.16038

[29] OpenAI, "What are tokens and how to count them?" https://help.openai.com/en/articles/4936856, 2024.

[30] J. Xie and H. Lee, "Prompt-based one-shot exact length-controlled generation with llms," *CoRR*, vol. abs/2508.13805, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.13805

[31] X. Zhang, Y. Chen, S. Hu, Z. Xu, J. Chen, M. K. Hao, X. Han, Z. L. Thai, S. Wang, Z. Liu *et al.*, "ınftybench: Extending long context evaluation beyond 100k tokens," pp. 15 262–15 277, 2024. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.814

[32] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[33] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, "Efficient streaming language models with attention sinks," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=NG7sS51zVF

[34] H. Jiang, Q. Wu, C. Lin, Y. Yang, and L. Qiu, "Llmlingua: Compressing prompts for accelerated inference of large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 13 358–13 376. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.825

[35] Y. Li, Y. Huang, B. Yang, B. Venkitesh, A. Locatelli, H. Ye, T. Cai, P. Lewis, and D. Chen, "Snapkv: LLM knows what you are looking for before generation," 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/28ab418242603e0f7323e54185d19bde-Abstract-Conference.html

[36] C. Packer, V. Fang, S. G. Patil, K. Lin, S. Wooders, and J. E. Gonzalez, "Memgpt: Towards llms as operating systems," *CoRR*, vol. abs/2310.08560, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2310.08560

[37] M. Kang, W. Chen, D. Han, H. A. Inan, L. Wutschitz, Y. Chen, R. Sim, and S. Rajmohan, "ACON: optimizing context compression for long-horizon LLM agents," *CoRR*, vol. abs/2510.00615, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.00615

[38] Y. Bai, J. Zhang, X. Lv, L. Zheng, S. Zhu, L. Hou, Y. Dong, J. Tang, and J. Li, "Longwriter: Unleashing 10, 000+ word generation from long context llms," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=kQ5s9Yh0WI

[39] W. Zhang, Z. Zhou, K. Wang, J. Fang, R. Xu, Y. Zhang, R. Wang, G. Zhang, X. Li, L. Sun *et al.*, "LIFEBENCH: evaluating length instruction following in large language models," 2025. [Online]. Available: http://papers.nips.cc/paper_files/paper/2025/hash/c84f00ccee3d35cee1901acb0e258dc7-Abstract-Datasets_and_Benchmarks_Track.html

[40] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. F. Christiano, "Recursively summarizing books with human feedback," *CoRR*, vol. abs/2109.10862, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2109.10862

[41] L. Ou and M. Lapata, "Context-aware hierarchical merging for long document summarization," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, ser. Findings of ACL, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., vol. ACL 2025. Association for Computational Linguistics, 2025, pp. 5534–5561. [Online]. Available: https://doi.org/10.18653/v1/2025.findings-acl.289

[42] Z. Zhou, C. Li, X. Chen, S. Wang, Y. Chao, Z. Li, H. Wang, Q. Shi, Z. Tan, X. Han *et al.*, "Llm×mapreduce: Simplified long-sequence processing using large language models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 27 664–27 678. [Online]. Available: https://doi.org/10.18653/v1/2025.acl-long.1341