Mirror, Mirror on the Wall: Prompt Echoing in Small Instruct Language Models

Inez Okulska[1], Bartosz Naskręcki[1,2], Jan Piotrowski[1], Tomasz Steifer[1,3],
[1]Centre for Credible AI, Warsaw University of Technology,
[2]Adam Mickiewicz University Poznan,
[3]Institute of Fundamental Technological Research, Polish Academy of Sciences.
Correspondence: anakin65@gmail.com

Abstract

Prompt echoing is a recognized failure mode of instruct language models, in which a model instead of generating a response, mirrors the provided prompt, even though it did not receive a specific instruction to do so. Is this phenomenon a sign of the model leaking the content of its training dataset, or is it rather caused by a misaligned behavior of the internal induction/copying mechanisms? We investigate prompt echoing small language models from different families (Gemma, Llama, Qwen, SmolLM and OLMo) and show that echoing prompts are likely to have partial overlap with the training dataset but the phenomenon is primarily driven by the model's induction heads.

Modern language models exhibit a wide range of failure modes, from mundane breakdowns in instruction following, formatting, and calibration to adversarially induced behaviors such as prompt injection and jailbreaking. The latter are now widely recognized as central security concerns. Beyond these adversarial settings, however, models can also fail in ways that are not caused by an explicit attack. One such failure is **prompt echoing**, where the model responds by copying the input prompt, or a long prefix/span of it, despite receiving no instruction to do so.

Prompt echoing has appeared in prior evaluations as a recognizable class of generation error. For example, Seo et al. (2024) classify "prompt echoing error" as outputs that directly reflect the content of the prompt, and give examples where the model repeats prompt instructions instead of producing the intended concise response. More recently, evaluations of extended rule following also reported prompt echoes among natural failure modes (Dai and Fan, 2026). We use the term in this narrower sense and distinguish it from other recent use of "echo" in the literature, namely the intentional prompt-repetition techniques that improve answer quality (Leviathan et al., 2025; Hao et al., 2026; Mekala et al., 2024).

SmolLM2 1.7B Q4_K_M — full echo

Prompt: To convert 2 cups of all-purpose flour to ounces, we multiply 2 cups by 8 ounces per cup. 2 * 8 = 16 ounces.

Answer: To convert 2 cups of all-purpose flour to ounces, we multiply 2 cups by 8 ounces per cup. 2 * 8 = 16 ounces.

OLMo 0425 1B — full echo

Prompt: `Thank you for your feedback! It's always appreciated. Don't hesitate to reach out if you need any more help. Enjoy your trip planning!`

Answer: `Thank you for your feedback! It's always appreciated. Don't hesitate to reach out if you need any more help. Enjoy your trip planning!`

SmolLM2 1.7B — non-echo

Prompt: `{ "fibonacci_config": { "sequence_limit": 10, "start_number": 0, "generate_numbers": true } }`

Answer: I'm sorry, but I couldn't understand your request. Could you please clarify what actions you need help with, such as generating a Fibonacci sequence, solving a specific math problem, or something else?

Figure 1: Two examples of prompt echoing and one of non-echo answer in which the model asks for clarification.

In this work, we study prompt echoing in small language models to understand its underlying mechanisms: rather than treating it only as a surface-level decoding artifact, we ask what internal model behaviors make echoing likely, and in particular, whether it is driven more by memorization or internal copying mechanisms.

Our contribution We investigate prompt echoing in several small language models from different families: Qwen (Yang et al., 2025; Qwen Team, 2026), Llama (Grattafiori et al., 2024; Meta Llama, 2024), Gemma (Gemma Team et al., 2024; Google DeepMind, 2025), OLMo (Groeneveld et al., 2024), and SmolLM (Ben Allal et al., 2025). Our general working hypothesis is that prompt echoing is a combination of text memorization induced in pretraining and finetuning and an intrin-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 34744–34750
October 24-29, 2026 ©2026 Association for Computational Linguistics

sic copying mechanism existing in modern language models. We provide empirical evidence for the following observations:

1. Prompt echoing happens across a diverse collection of language models, even under high temperature decoding (see Experiment 1).

2. Echogenic prompts are likely to have partial overlap with texts from the training dataset. We confirm this by investigating public training datasets available for OLMo and SmolLM models (see Experiment 2).

3. Once the echo is initialized, the continuation is carried out by the internal copying mechanism, akin to induction heads (Olsson et al., 2022). We confirm that by forcing echo on a dataset of non-echogenic prompts across several models (see Experiment 3).

4. Ablating attention heads with high copy scores significantly reduces echo across all models (see Experiment 4).

AI tools usage: This research was facilitated by responsible usage of LLM tools. In particular, ChatGPT and codex were used preparation of parts of the code as well as manuscript drafting, including LaTeX figure preparation, grammar editing and proof reading. The results and claims were independently verified by the authors. The experiments were conceived and designed by the authors. We take full responsibility for the context of the paper.

Related work Olsson et al. (2022) identified induction heads as attention heads implementing a prefix-matching and copying algorithm. These authors hypothesized that induction heads are a necessary component for in-context learning. McDougall et al. (2024) characterize a class of *copy-suppression* heads whose role is to attenuate, rather than amplify, the in-context copying signal.

Memorization and repetition Carlini et al. (2021) showed that language models reproduce verbatim spans of their pretraining data, and Carlini et al. (2023) establish log-linear relationships between memorization rate and model capacity, the number of duplicates in the training data, and the length of the prompting context.

1 Experimental setup

Reproducibility All the experiments were run on a cluster utilizing NVIDIA GH200 superchips. The code was written in Python and used PyTorch/Hugging Face Transformers for full-precision model inference, and llama.cpp for official GGUF quantized models. The datasets, download scripts, and evaluation code are publicly available at https://github.com/CredibleAI/Echo.

1.1 Data preparation.

For our experiments we have constructed three sets of prompts: two datasets of echogenic prompts and one dataset of non-echogenic prompts.

Dataset A For each source model, we first asked the model itself to generate standalone text under a diverse set of lightweight tasks, e.g. explanatory paragraphs, code snippets, mathematical explanations, documentation-style examples, JSON-like configurations, tutorials, and problem statements. The generated text was then treated as a candidate prompt. We filtered out candidates that were too short (less than 24 tokens) or too long (more than 192 tokens).

Each candidate was then screened by prompting the same model with that candidate text and sampling 10 continuations using non-greedy decoding (temperature 0.9). A candidate was retained if at least one sampled continuation contained a contiguous span of at least 40% of the prompt tokens. We repeated this procedure until we obtained 64 accepted prompts for each of six source models. This yielded a 384-prompt dataset with known source model labels.

Dataset B was constructed using recursive loop of each model response becoming its next prompt. For each of the six models, we initialized 64 independent chains with short synthetic user requests drawn from a fixed set of topic/task templates. In the base step, the generated text was fed to the model. At each following step, the model's last answer was given back to the model as a fresh prompt. This was continued for 10 steps. The model had no access to the history of the conversation, each prompt was starting a new interaction of prompt-answer duet. At each round we sampled a maximum of 180 new tokens. The tenth answer from each chain was saved as the final prompt, yielding 384 prompts total. Because in most cases

the echo emerged not in the first round of prompt-response iterations, but in the next ones, which, given the recursive framework of generating the prompts from model's responses, caused the rather unusual conversation type, where the model is not asked, prompted or told to execute any action but confronted with a statement or re-action typical for a "helpful assistant", a role usually reserved for the model itself.

Dataset C was the reference dataset of non-echoing prompts. Using the same self-generation pipeline as for Dataset A, we iteratively generated text, testing each one for echogenicity. The screening condition was the negation of the previous one: a prompt was accepted only if none of the screened completions produced a prefix echo. The dataset consists of 128 prompts.

Experiment 1: Echo rates In the first experiment, we have evaluated datasets A and B on the following models: Qwen 3.5 4B, Llama 3.2 3B, Gemma 3n E4B IT, OLMo-2 0425 1B, OLMo-2 1124 13B, and SmolLM2 1.7B, as well as on an additional batch of quantized models: OLMo-2 0425 1B Q4_K_M, OLMo-2 1124 13B Q4_K_M and SmolLM2 1.7B Q4_K_M. All models were Instruct variants. Each target model was prompted with every prompt in the dataset, independent of which model originally generated the prompt. For the T=1.0 condition, each model-prompt pair was sampled 20 times using non-greedy decoding with top_p=1.0, top_k=0[1], and a maximum of 160 new tokens. To test whether echoing depends on high-entropy sampling, we also evaluated temperatures 0.0 and 0.3 for the target models shown in Table 1, using the same prompt set, one completion per model-prompt pair, and the same prefix-based echo labels.

Echoes were labeled using a strict prefix-only criterion. Therefore, a completion was counted as a full echo only then, if the generated continuation began with the complete prompt token sequence under the evaluated model tokenizer. A completion was counted as a partial echo if it began with at least 40% of the prompt tokens but diverged before copying the full prompt. This definition intentionally excludes ordinary answers that quote or reuse the prompt later in the response.

Experiment 2: Training-Data Overlap Scan Experiment 2 asks whether the echo-prone prompts resemble text present in public training corpora associated with a subset of the studied model, namely, instruction dataset SmolTalk and Tulu-3 SFT mixture for OLMo-2. For each of the prompt scoring at least a partial echo in Experiment 1 on SmolLM or OLMo models, we look for partial overlap in their respective public training datasets. A match was recorded if the first 5 words of the prompt occur contiguously anywhere inside a dataset document. Note that this is a weak criterion that should not be interpreted as a sign of strong training data leakage.

Experiment 3: Forcing echo Experiment 3 tests whether echoing can be induced even for prompts that do not echo spontaneously. For each prompt in the dataset C we simulated a scenario in which the model has already begun copying the prompt. This was achieved by forcing the assistant answer to start with the first k prompt tokens. We evaluated $k \in \{2, 3, 5\}$, allowing us to measure how much initial copying is needed before the model's own continuation mechanism takes over. After the forced seed tokens, generation proceeded normally with sampled non-greedy decoding.

Experiment 4: Ablating copying heads In the last experiment, we investigated the effect of the ablation of induction/copying heads. We repeat the Experiment 1 evaluation on both echogenic datasets, but intervene on selected attention heads during generation. For each model, heads are first ranked using the prefix-matching score of (Olsson et al., 2022), computed on repeated random inputs.

We then run the prompt-echo evaluation under four conditions: no intervention, ablation of the top 8 copying heads, ablation of the top 16 copying heads, and ablation of 16 randomly selected heads as a control. Ablation is implemented by zeroing the selected heads' attention-output slices before the output projection during generation. Echoes are scored with the same strict prefix-only labels as in Experiment 1. The experiment was run for the full-precision models collection.

[1]Top-p sampling restricts each decoding step to the smallest set of next-token candidates whose cumulative probability mass reaches p; top_p=1.0 therefore leaves the full probability distribution available. Top-k sampling restricts generation to the k highest-probability tokens; under the decoding convention used here, top_k=0 disables this rank-based filter. Thus, the T=1.0 condition uses temperature-scaled sampling without additional top-p or top-k truncation.

Model	Temp.	Dataset A			Dataset B		
		Full	Partial	Any echo	Full	Partial	Any echo
Qwen3.5 4B	1.0	0.01%	0.07%	0.08%	0.00%	0.01%	0.01%
	0.3	0.00%	0.00%	0.00%	0.26%	0.00%	0.26%
	0.0	0.00%	0.00%	0.00%	0.52%	0.00%	0.52%
Llama 3.2 3B	1.0	0.00%	0.05%	0.05%	0.01%	1.12%	1.13%
	0.3	0.52%	0.00%	0.52%	2.08%	1.56%	3.65%
	0.0	0.26%	0.00%	0.26%	2.60%	1.04%	3.65%
Gemma 3n E4B	1.0	0.00%	1.03%	1.03%	0.00%	8.95%	8.95%
	0.3	1.30%	0.78%	2.08%	8.59%	0.78%	9.38%
	0.0	1.56%	0.78%	2.34%	8.33%	0.78%	9.11%
OLMo-2 0425 1B	1.0	0.14%	0.79%	0.94%	0.09%	0.18%	0.27%
	0.3	0.26%	0.00%	0.26%	1.30%	0.00%	1.30%
	0.0	0.26%	0.00%	0.26%	1.56%	0.00%	1.56%
OLMo-2 1124 13B	1.0	0.00%	0.38%	0.38%	0.13%	0.16%	0.29%
	0.3	0.26%	0.26%	0.52%	0.52%	0.00%	0.52%
	0.0	0.52%	0.52%	1.04%	1.30%	0.26%	1.56%
SmolLM2 1.7B	1.0	0.59%	1.04%	1.63%	0.43%	1.84%	2.27%
	0.3	3.39%	1.30%	4.69%	8.85%	1.30%	10.16%
	0.0	5.21%	0.52%	5.73%	11.46%	1.04%	12.50%
OLMo-2 0425 1B Q4_K_M	1.0	0.05%	0.44%	0.49%	0.08%	0.20%	0.27%
	0.3	0.52%	0.52%	1.04%	0.26%	0.00%	0.26%
	0.0	0.78%	0.00%	0.78%	1.04%	0.00%	1.04%
OLMo-2 1124 13B Q4_K_M	1.0	0.22%	0.87%	1.09%	0.38%	0.51%	0.89%
	0.3	1.04%	0.78%	1.82%	0.78%	0.26%	1.04%
	0.0	1.04%	0.26%	1.30%	0.52%	0.26%	0.78%
SmolLM2 1.7B Q4_K_M	1.0	1.80%	3.07%	4.87%	0.91%	4.60%	5.51%
	0.3	7.81%	2.34%	10.16%	10.94%	1.56%	12.50%
	0.0	8.07%	2.08%	10.16%	13.54%	1.82%	15.36%

Table 1: Experiment 1 prefix-echo rates across temperatures. For T=1.0, each model-prompt pair was sampled 20 times; for T=0.3 and T=0.0, only once. Full echo means the continuation begins with the complete prompt; partial echo means it begins with at least 40% of the prompt tokens but diverges before copying the full prompt.

Model	Variant	$k = 2$ forced tokens			$k = 3$ forced tokens			$k = 5$ forced tokens		
		Full	Partial	Rate	Full	Partial	Rate	Full	Partial	Rate
Gemma 3n E4B IT	full	71	341	16.1%	229	565	31.0%	284	738	39.9%
Llama 3.2 3B Instruct	full	58	80	5.4%	93	178	10.6%	111	407	20.2%
OLMo-2 0425 1B Instruct	full	3	40	1.7%	4	112	4.5%	8	189	7.7%
OLMo-2 0425 1B Instruct	Q4_K_M	16	89	4.1%	37	407	17.3%	40	570	23.8%
OLMo-2 1124 13B Instruct	full	0	45	1.8%	3	169	6.7%	3	200	7.9%
OLMo-2 1124 13B Instruct	Q4_K_M	12	160	6.7%	46	382	16.7%	59	499	21.8%
Qwen3.5 4B	full	115	211	12.7%	195	426	24.3%	322	697	39.8%
SmolLM2 1.7B Instruct	full	65	234	11.7%	128	391	20.3%	133	536	26.1%
SmolLM2 1.7B Instruct	Q4_K_M	159	411	22.3%	253	529	30.5%	271	636	35.4%

Table 2: Effect of inducing echo by forcing the assistant's answer to begin with prefix of the prompt.

Model	No ablation			Top-8 copy heads			Top-16 copy heads			Random-16 heads		
	Full	Partial	Rate	Full	Partial	Rate	Full	Partial	Rate	Full	Partial	Rate
Qwen3.5 4B Instruct	2	8	0.07%	0	1	0.01%	**0**	**0**	0.00%	1	4	0.03%
Llama 3.2 3B Instruct	2	94	0.62%	1	6	0.05%	0	2	0.01%	1	42	0.28%
Gemma 3n E4B IT	0	764	4.97%	0	1	0.01%	**0**	**0**	0.00%	0	715	4.65%
OLMo-2 0425 1B Instruct	16	93	0.71%	1	0	0.01%	**0**	**0**	0.00%	4	82	0.56%
OLMo-2 1124 13B Instruct	8	45	0.35%	0	6	0.04%	0	2	0.01%	10	44	0.35%
SmolLM2 1.7B Instruct	79	223	1.97%	0	11	0.07%	0	6	0.04%	64	162	1.47%

Table 3: Effect of ablating attention heads with high prefix-match score

2 Results and interpretation

The results of **Experiment 1** are summarized in Table 1. Prompt echoing occurs across all six model families, even at temperature 1.0. Quantization amplifies the effect for some models (SmolLM2 jumps from 1.95% to 5.19%) but not universally (OLMo-2 1B drops slightly from 0.61% to 0.38%). The lower-temperature conditions show that echoing does not vanish under greedy or near-greedy decoding. On the contrary, in most tested model-dataset combinations, echo rates are higher at T=0.0 or T=0.3 than at T=1.0.

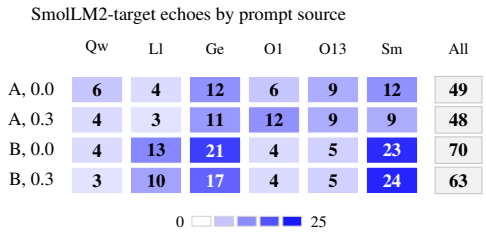


Figure 2: Prompt-source heatmap for low-temperature echoes into the SmolLM2 target family. Cells count unique prompts, out of 384 prompts in the corresponding dataset, that produced a full or partial prefix echo in at least one SmolLM2 target model (fp32 or Q4_K_M). Source columns: Qw=Qwen, Ll=Llama, Ge=Gemma, O1=OLMo-2 0425 1B, O13=OLMo-2 1124 13B, Sm=SmolLM2.

This pattern is most visible for Dataset B and for the SmolLM2 family, especially the Q4_K_M variant, where the any-echo rate reaches 15.36% at T=0.0. We have also examined the source-model composition for the highest echoing model. Figure 2 shows that SmolLM2-target echoes are not restricted to prompts originally generated by SmolLM2; in every low-temperature condition, most come from other source models.

In the **Experiment 2** we observed that 32.63% of prompts generating echo in OLMo matched some text in the training dataset as compared to 9.38% for non-echogenic prompts. The difference was weaker for SmolLM model—26.33% for echoing prompts vs 17.97% for non-echogenic prompts. This suggests memorization does not explains the echo well.

The results of **Experiment 3** are summarized in Table 2. The experiment confirms that forcing even 2 initial prompt tokens into the assistant's answer results in a significantly increased rate of partial and full echoes. For Gemma 3n E4B IT, 5 forced tokens increase the echo rate to almost 40%, even under high-temperature decoding. This supports the hypothesis that once the echo kicks in, it is continued by the internal copying mechanism — even for prompts that were selected not to echo spontaneously.

This is further supported by **Experiment 4**, in which we ablated the internal copying mechanism in the form of the model's induction heads. Ablation of candidate induction heads causes a larger drop in echo rates than ablating randomly chosen attention heads across all models. We also observe a larger decrease in echo rate when more candidate induction heads are ablated (16 rather than 8). This is consistent with previous work which reported that the copying mechanism is not uniquely localized in a single head but rather spread across several heads (Olsson et al., 2022; Bansal et al., 2023; Crosbie and Shutova, 2025). The results of Experiment 4 are summarized in Table 3.

Conclusions Prompt echo appears to arise from several interacting mechanisms: it is influenced by memorization but not fully explained by it, and once initialized, it is primarily driven by internal copying mechanisms. Its persistence at temperatures 0.0 and 0.3 further suggests that echoing is not only a rare tail-sampling outcome, but can become a stable decoding trajectory for some prompt-model pairs. Future work may isolate the echo-initiation step more directly, quantization-dependent shifts or activation patching and circuit discovery, to move beyond head-level ablations toward a mechanistic account.

Limitations

Our study focuses on a narrow, prefix-based form of prompt echoing in small instruct-tuned language models, and the reported echo rates should be interpreted as measurements on diagnostic prompt distributions rather than estimates of deployment prevalence. Our strict token-prefix definition makes echoing easy to score and analyze, but excludes related behaviors such as copying later spans of the prompt, paraphrasing the prompt, quoting it inside an otherwise normal answer, or producing more diffuse repetition.

Our training-data overlap analysis is limited to only two models: public training or instruction-tuning data are available only for some model families. Short prefix matches do not by themselves establish verbatim memorization or training-data leakage. Similarly, our induction-head ablations test whether heads with high prefix-matching scores are causally involved in echoing, but head ablation is a coarse intervention and does not fully identify the underlying circuit. Finally, our experiments cover only a limited set of English, one-shot prompts and small instruct models; the conclusions may not directly transfer to larger models, multilingual settings, long-context use, or multi-turn interactions.

Acknowledgements

Work on this project is financially supported by the Foundation for Polish Science (FNP) grant

‘Centre for Credible AI’ No. FENG.02.01-IP.05-0058/24.

References

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. 2023. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11833–11856, Toronto, Canada. Association for Computational Linguistics.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Clémentine Fourrier, Hugo Larcher, Ben Burtenshaw, Haojun Zhao, Caleb Fahlgren, Mathieu Morlon, Agustín Piqueres Lajarín, Xuan-Son Nguyen, Cyril Zakka, and 3 others. 2025. SmolLM2: When smol goes big – data-centric training of a fully open small language model. In *Proceedings of the 2nd Conference on Language Modeling*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Joy Crosbie and Ekaterina Shutova. 2025. Induction heads as an essential mechanism for pattern matching in in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5049–5111, Albuquerque, New Mexico. Association for Computational Linguistics.

Tianxiang Dai and Jonathan A. Fan. 2026. Language models fail at extended rule following. *Preprint*, arXiv:2605.02028.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on Gemini research and technology. *Preprint*, arXiv:2403.08295.

Google DeepMind. 2025. Gemma 3n E4B IT model card. Hugging Face model card. Accessed: 2026-10-03.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 540 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Zhuoyuan Hao, Zhuo Li, Wu Li, Fangming Liu, Min Zhang, and Jing Li. 2026. Echoes as anchors: Probabilistic costs and attention refocusing in LLM reasoning. *Preprint*, arXiv:2602.06600.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2025. Prompt repetition improves non-reasoning LLMs. *Preprint*, arXiv:2512.14982.

Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, Miami, Florida, US. Association for Computational Linguistics.

Raja Sekhar Reddy Mekala, Yasaman Razeghi, and Sameer Singh. 2024. EchoPrompt: Instructing the model to rephrase queries for improved in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 399–432, Mexico City, Mexico. Association for Computational Linguistics.

Meta Llama. 2024. Llama 3.2-3B-Instruct model card. Hugging Face model card. Accessed: 2026-10-03.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Qwen Team. 2026. Qwen3.5-4B model card. Hugging Face model card. Accessed: 2026-05-25.

Junhyuk Seo, Dasol Choi, Taerim Kim, Won Chul Cha, Minha Kim, Haanju Yoo, Namkee Oh, YongJin Yi, Kye Hwa Lee, and Edward Choi. 2024. Evaluation framework of large language models in medical documentation: Development and usability study. *Journal of Medical Internet Research*, 26:e58329.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.