Analysis of Emergence of Reasoning in Language Models: Factors, Thresholds and Interpretations

Yen-Che Hsiao
Department of Electrical and Computer Engineering
University of Connecticut
Storrs CT 06269, USA
yen-che.hsiao@uconn.edu

Abhishek Dutta
Department of Electrical and Computer Engineering
University of Connecticut
Storrs CT 06269, USA

Abstract

This work investigates and provides insights into the reasoning thresholds of open-source, decoder-only, transformer-based language models (LMs) with less than three billion parameters by studying three key aspects: reasoning with in-context learning, zero-shot reasoning, and fine-tuning of small models for zero-shot reasoning. The reasoning ability of LMs using in-context learning is evaluated using deductive reasoning tasks, where we show that reasoning ability is influenced by model size and architecture, such as feedforward width and number of attention heads, as well as by properties of the pretraining data, including scale, diversity, long-range coherence, and the ordering of in-context demonstrations. For zero-shot reasoning, we show that fine-tuning LMs on instruction and code data, the use of prompting strategies such as plan-and-solve and role-play, and the depth of LMs can all contribute to improved zero-shot reasoning performance. Regarding the fine-tuning of small LMs, we show that LMs can acquire logical reasoning abilities through instruction tuning with chain-of-thought data, with or without exemplars, and through knowledge distillation. To support the above conclusions, we analyze multi-head attention to correlate with multiple reasoning paths and apply attention unembedding to identify which tokens are written to the residual stream. These findings provide a clearer understanding of the conditions under which reasoning abilities emerge in LMs.

1 Introduction

In-context learning (ICL) has emerged as a powerful capability in language models (LMs), where models perform tasks by conditioning on examples provided in the prompt without any parameter updates [1, 2]. [3] demonstrated that scaling model size from 0.1B to 175B in GPT-3 leads to consistent performance gains across a wide range of benchmarks under ICL.

Building on this, [4] introduced chain-of-thought (CoT) prompting, which improves ICL performance by eliciting intermediate reasoning steps. Their findings on large-scale proprietary models such as PaLM 540B [5], LaMDA [6], and GPT-3 [3] showed that CoT can significantly enhance reasoning accuracy. However, these investigations have largely focused on a narrow set of reasoning tasks or on very large, closed-source models. As a result, the reasoning capabilities of smaller open-source LMs, particularly those with fewer than 3 billion parameters, remain poorly understood.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

To address this gap, we systematically evaluate the ICL reasoning ability and zero-shot reasoning ability of 29 decoder-only transformer-based LMs with fewer than 3 billion parameters across commonsense, mathematical, and deductive reasoning tasks. We aim to identify the key factors that affect reasoning performance in this parameter regime. Recent work has examined individual aspects of ICL reasoning, such as multilingual reasoning ability [7], the effect of irrelevant rationales [8, 9], and the structure of prompts and training data [10, 9]. Zero-shot prompting strategies including *Let's think step-by-step* [11], *Plan-and-Solve* [12], and *Role-Play* have also been explored in larger LMs, but systematic evaluations over a wide range of small open-source models are still lacking.

Unlike [7], we focus exclusively on English-language models. In contrast to [8] and [9], we avoid adding irrelevant rationales to the prompt, since models in this scale often fail on clean inputs and additional noise could obscure meaningful trends. Motivated by findings that structured training data influences ICL behavior [9], we also investigate whether supervised fine-tuning can improve reasoning performance in models smaller than 0.5B parameters.

Our work is similar to [13], which studies the ICL and zero-shot reasoning ability of various LMs on tasks such as math, commonsense, and sorting. However, we extend this line of work in several directions. First, we provide deeper analysis into the architectural and pretraining factors that affect ICL and zero-shot reasoning. Second, we evaluate zero-shot reasoning across 29 models grouped by size and instruction tuning, using a variety of prompting strategies. Third, and uniquely, we test the ICL reasoning ability of LMs on a synthetic deductive reasoning dataset. This dataset allows us to vary the number of in-context exemplars and the logical complexity of the task. Finally, we fine-tune small LMs and show that zero-shot reasoning capabilities can be acquired through fine-tuning or transferred via knowledge distillation.

To support our conclusions, we conduct attention-based interpretability analysis. We examine whether multi-head attention correlates with multiple reasoning paths and apply attention unembedding to identify which tokens are written to the residual stream during inference. These findings offer a clearer understanding of the conditions under which reasoning abilities emerge in LMs, especially in small, open-source models.

2 Approaches

2.1 Language models

We evaluate 29 open-source decoder-only transformer-based language models with parameter counts ranging from 135 million to 3.21 billion. These include models from SmolLM2 [14], Gemma2 [15], Gemma3 [16], Qwen2.5 [17], Qwen3 [18], Deepseek-R1 [19], Llama3.2 [20], and OLMo2 [21]. Instruction-tuned variants are included where available. Further architectural and training details are provided in the Appendix.

2.2 Evaluating ICL Reasoning Ability on the PrOntoQA-OOD Dataset

The PrOntoQA dataset [22] is a synthetic and programmable dataset designed for evaluating deductive reasoning in LMs. The PrOntoQA-OOD extension [23] broadens the evaluation by incorporating compositional proofs and a complete set of logical deduction rules.

Each example contains a set of premises, a conclusion to be proven or disproven, and a gold CoT proof. The dataset supports six deductive rules: implication elimination (modus ponens), conjunction introduction, conjunction elimination, disjunction introduction, disjunction elimination (proof by cases), and proof by contradiction.

2.2.1 Prompt Construction for ICL Reasoning Evaluation

To test ICL reasoning, we format prompts by concatenating multiple exemplars followed by a test question. Each exemplar begins with "Q: ", followed by premises, continues with ". Prove: ", then the conclusion, and ends with "\ nA: " and the corresponding gold CoT. Exemplars are separated by two newline characters. The test instance is appended in the same format, excluding the gold CoT.

Exemplars are dynamically generated per test question using the official dataset generation code [22], ensuring variation in exemplars across test cases. Each model is evaluated across configurations with

1 to 10 exemplars. For each configuration, we test all six deductive rules. For implication elimination, conjunction introduction, conjunction elimination, and disjunction introduction, reasoning hop is varied from one to five. Disjunction elimination and proof by contradiction are tested only with one-hop reasoning due to limitations in the released code.

2.2.2 Parsing and Evaluation of LM Outputs

We isolate the CoT response by removing the prompt from the model output. The correctness of each CoT is evaluated using the "analyze_results.py" script provided by PrOntoQA-OOD [23], which verifies logical consistency with the corresponding deduction rule.

2.2.3 Attention Analysis for ICL Reasoning Attribution

To interpret internal model behavior during ICL reasoning, we analyze the attention layers in the LMs. Given matrices $Q, K, V \in \mathbb{R}^{m \times d_{\text{model}}}$, the attention output of head $i = 1, 2, \ldots, h$ is computed as:

$$H_i = \text{softmax}\left(M + \frac{QW_i^Q (KW_i^K)^\top}{\sqrt{d_{\text{model}}}}\right) VW_i^V, \tag{1}$$

where $W_i^Q, W_i^K \in \mathbb{R}^{d_{\text{model}} \times r}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times q}$, $h \in \mathbb{N}$ is the number of heads, $m \in \mathbb{N}$ is the context length, $d_{\text{model}} \in \mathbb{N}$ is the embedding dimension, and $r = q = d_{\text{model}}/h$. The causal mask $M \in \mathbb{R}^{m \times m}$ enforces autoregressive constraints. The output of the masked multi-head self-attention is:

$$H = [H_1 \; H_2 \; \ldots \; H_h] W^O, \tag{2}$$

where $W^O \in \mathbb{R}^{qh \times d_{\text{model}}}$.

We analyze the final row of the attention map at each layer to investigate how attention mechanisms correlate multiple reasoning paths. Specifically, we compute the attention map for each head using

$$B = \text{softmax}\left(M + \frac{QW_i^Q (KW_i^K)^\top}{\sqrt{d_{\text{model}}}}\right), \tag{3}$$

and extract the last row of B for each head at each layer. For each attention map, we identify the maximum attention score that corresponds to the correct next token.

To determine which tokens the head writes to the residual stream, we apply attention unembedding to the normalized output of each attention layer as in [24]. Let H denote the attention output at a given layer. We project this output to the vocabulary space using

$$Y = \text{LayerNorm}(H) W^{\text{last}}, \tag{4}$$

where $W^{\text{last}} \in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}$ is the unembedding matrix and $|\mathcal{V}|$ is the vocabulary size.

2.2.4 Fine-Tuning Small LMs on ICL Reasoning Tasks

To assess how fine-tuning affects ICL reasoning, we fine-tuned two small-scale LMs: SmolLM2-135M and SmolLM2-135M-Instruct. The training set contains 1800 exemplars (300 per rule × 6 rules), drawn from PrOntoQA-OOD and split into 90% training and 10% validation.

We fine-tune with causal language modeling using the Hugging Face Trainer API on the last two layers for each LM. Training uses the AdamW optimizer with weight decay 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ [25], for 100 epochs, batch size 1, and learning rate 2×10^{-5}. The best-performing checkpoint is selected based on validation loss.

2.3 Zero-Shot Evaluation of Commonsense and Mathematical Reasoning

2.3.1 CommonsenseQA Dataset

CommonsenseQA [26] includes 12,247 multiple-choice questions based on ConceptNet [27]. We evaluate models on the 1,221-question validation set. Each prompt is structured as "Q: " followed by the question, then "Answer Choices: (A)" and the subsequent options, following the template in [4].

2.3.2 GSM8K Dataset

GSM8K [28] contains 1,319 grade-school math word problems requiring multi-step reasoning. Compared to complex datasets like MATH [29], GSM8K is more suitable for evaluating small LMs for our experiment.

2.3.3 Zero-Shot Prompting Strategies

We evaluate models under three zero-shot prompting methods: "Let's think step-by-step" [11], Plan-and-Solve [12], and Role-Play [30]. Role-Play prompting [30] is applied only to instruction-tuned models, as its format presumes system-role awareness.

Answer extraction follows [12]. For CommonsenseQA, the answer trigger is: "Therefore, among A through E, the answer is most likely ". We extract either the letter in parentheses (e.g., "(C)") or the first letter. If the extracted letter is not in A–E, the model is considered as nonresponsive.

For GSM8K, the answer trigger is: "Therefore, the answer (arabic numerals) is most likely ". We extract the first integer after the trigger. If no integer appears, the model is considered not to have answered.

2.3.4 Fine-Tuning on CommonsenseQA

We fine-tuned the last-two layer of two LMs: SmolLM2-135M and SmolLM2-135M-Instruct. To construct the training and validation sets, we used the gemma-2-9b-it model to generate CoT by prompting it with the training set of the CommonsenseQA dataset [26] and the same CoT prompt format used in [4]. We retained only the generated samples where the predicted answer matched the ground-truth answer. In total, we collected 1,000 question–answer pairs, with 900 used for training and 100 for validation. Each sample contains one question and its corresponding CoT answer, and we set the context length to the maximum number of tokens across all 1,000 samples.

All models are trained for 100 epochs using AdamW with a learning rate of 2×10^{-5}, weight decay of 0.01, batch size of 1, and a linear learning rate schedule.

2.3.5 Fine-Tuning on GSM8K

Using the same setup as in Section 2.3.4, we fine-tuned the last two layers of the same two LMs on the 1,319-sample GSM8K test set [28]. Each training sample consists of a question and a final answer from [28], with all "\ n#### " markers replaced by "The answer is ".

3 Results and discussion

3.1 Results and Discussion on the ICL Reasoning using the PrOntoQA-OOD Dataset

We evaluated the ICL reasoning ability of 29 LMs using the procedure described in Section 2.2.1, applied to the PrOntoQA-OOD dataset [23]. Each model was tested on four deductive rules with 1 to 5 reasoning hops, disjunction elimination and proof by contradiction with 1 hop, using 1- to 10-shot exemplars. For each combination of shot count, reasoning depth, and task type, 10 proofs were used to assess performance. The results, shown in Figure 1, reveal that accuracy generally increases from 1 to 4 shots and flattens thereafter. Group-wise analysis indicates that higher values in parameter count, embedding dimension, feedforward width, number of attention heads, model depth, and context length, as well as pretraining on long-range data, correlate with improved ICL reasoning performance. In contrast, instruction tuning does not yield a consistent benefit.

These empirical trends are consistent with prior findings suggesting that ICL is an emergent behavior in transformer-based models. The ability to perform a task via few-shot prompting is emergent when a model initially exhibits random performance until a critical scale is reached, after which performance increases to well above random [31]. Specialized prompting or fine-tuning methods can also be emergent in the sense that they only become effective beyond a certain model scale.

ICL enables a model to perform tasks by conditioning on input-output examples without any parameter updates. It achieves this by leveraging the prompt to retrieve and recombine latent concepts acquired during pretraining [32]. The efficacy of ICL is closely tied to the pre-training phase (domain

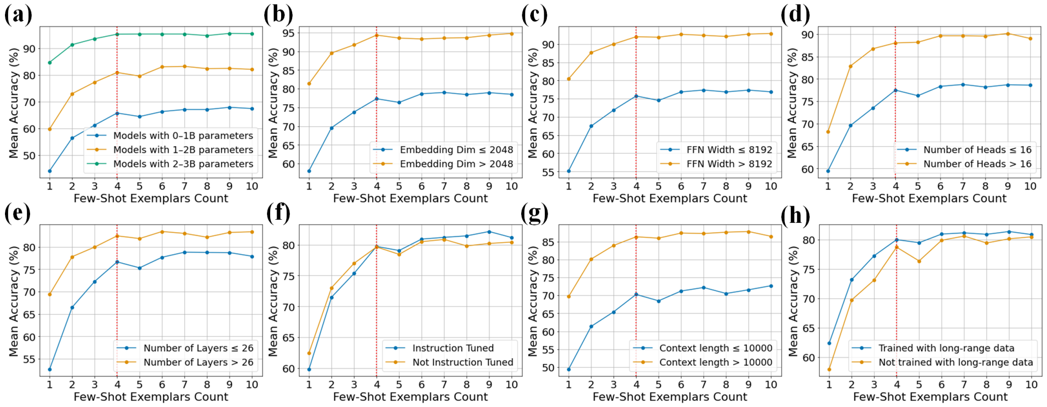


Figure 1: Averaged accuracy of 29 LMs evaluated on the PrOntoQA-OOD dataset using 1- to 10-shot exemplars. Each model is tested on disjunction elimination with 7 reasoning hops, proof by contradiction with 13 hops, and four deductive rules with 1 to 5 reasoning hops. For each combination of shot count, reasoning hop, and task type, 10 proofs are used to assess ICL reasoning ability. Accuracy generally increases from 1 to 4 shots and flattens thereafter, as indicated by the vertical red dotted line. (a) Models grouped by parameter count: 0–1B (blue), 1–2B (orange), 2–3B (green). (b) By embedding dimension: > 2048 (orange) vs. ≤ 2048 (blue). (c) By feedforward width (the number of hidden neurons): > 8192 (orange) vs. ≤ 8192 (blue). (d) By number of attention heads: > 16 (orange) vs. ≤ 16 (blue). (e) By model depth: > 26 layers (orange) vs. ≤ 26 (blue). (f) Instruction-tuned (blue) vs. non-instruction-tuned models (orange). (g) By context length: > 10000 (orange) vs. ≤ 10000 (blue). (h) Pretrained on long-range data (blue) vs. not pretrained on long-range data (orange). Overall, higher values in parameter count, embedding dimension, feedforward width, number of attention heads, number of layers, and context length, as well as pretraining on long-range data, correlate with higher mean accuracy in ICL reasoning. In contrast, instruction tuning does not exhibit a consistent benefit.

specificity) and the scale of model parameters. During pre-training, models acquire a broad range of semantic prior knowledge from the training data, which later aids task-specific learning representation. This process can be formalized as

$$p(\text{output} \mid \text{prompt}) = \int_{\text{concept}} p(\text{output} \mid \text{concept}, \text{prompt})\, p(\text{concept} \mid \text{prompt})\, d(\text{concept}).$$

ICL is an emergent phenomenon, as their transformer model moves beyond memorization of the pretraining tasks when there is sufficient diversity and scale in pre-training data [33]. ICL can emerge when pretraining documents have long-range coherence and the LM develops a broad set of skills and pattern recognition abilities [3].

Semantically unrelated labels directs the model to learn the input-label mappings from scratch, as it can no longer rely on its semantic priors for task completion [34]. Larger models are more adept at this form of ICL over smaller ones, indicating their ability to adapt to new task descriptions without relying solely on pre-trained semantic knowledge.

At a mechanistic level, ICL in attention-only transformers is facilitated by head composition. For example, two-layer models can form induction heads that search the prompt for previous occurrences of a given token and use this information to infer patterns [35]. These heads are essential for aligning current input with past context during few-shot reasoning.

Finally, LMs can be viewed as meta-optimizers, with ICL functioning as implicit fine-tuning. Recent theoretical work shows a duality between transformer attention and gradient descent. In particular, a transformer with K layers and in-context demonstrations can approximate K steps of gradient descent on training data [1].

To support the above conclusions, we use multi-head attention scores to correlate the attention heads with multiple reasoning paths, as illustrated in Figure 2. We test the Qwen2.5-0.5B model on a conjunction introduction task using one exemplar. The model is prompted with an initial input and then incrementally extended by concatenating one token at a time from the expected answer. At each step, we extract the last row of the attention map from every head at every layer and record the maximum attention weight corresponding to the correct next token.

These values are visualized as heat maps, where each map represents a reasoning step and highlights which attention heads are attending most strongly to the correct token. Notably, similar patterns appear in the heat maps at steps 1, 2, 5, 6, 14, 15, 18, 19, and 20, as well as in steps 3, 7, and 12. This suggests that certain heads across layers may specialize in supporting particular reasoning patterns, depending on the context or stage of the task.

To determine which token attention heads contributes to in the residual stream, we perform attention unembedding on the output of each attention layer. Specifically, for each layer, we project the attention output into the vocabulary space and identify the token with the highest probability. We apply this procedure to a conjunction elimination task with one exemplar, where the correct next token is "moderate," as shown in Figure 3. The analysis is conducted on three different language models: LLaMA-3.2-3B, Qwen3-1.7B, and SmolLM2-360M, all of which correctly predict the next token using greedy decoding.

The line plots show that the probability of the correct next token increases in the later layers, indicating that deeper attention layers are more aligned with the final prediction. This suggests that later attention layers play a more significant role in writing the correct token to the residual stream. Results for the other LMs are provided in the Appendix.

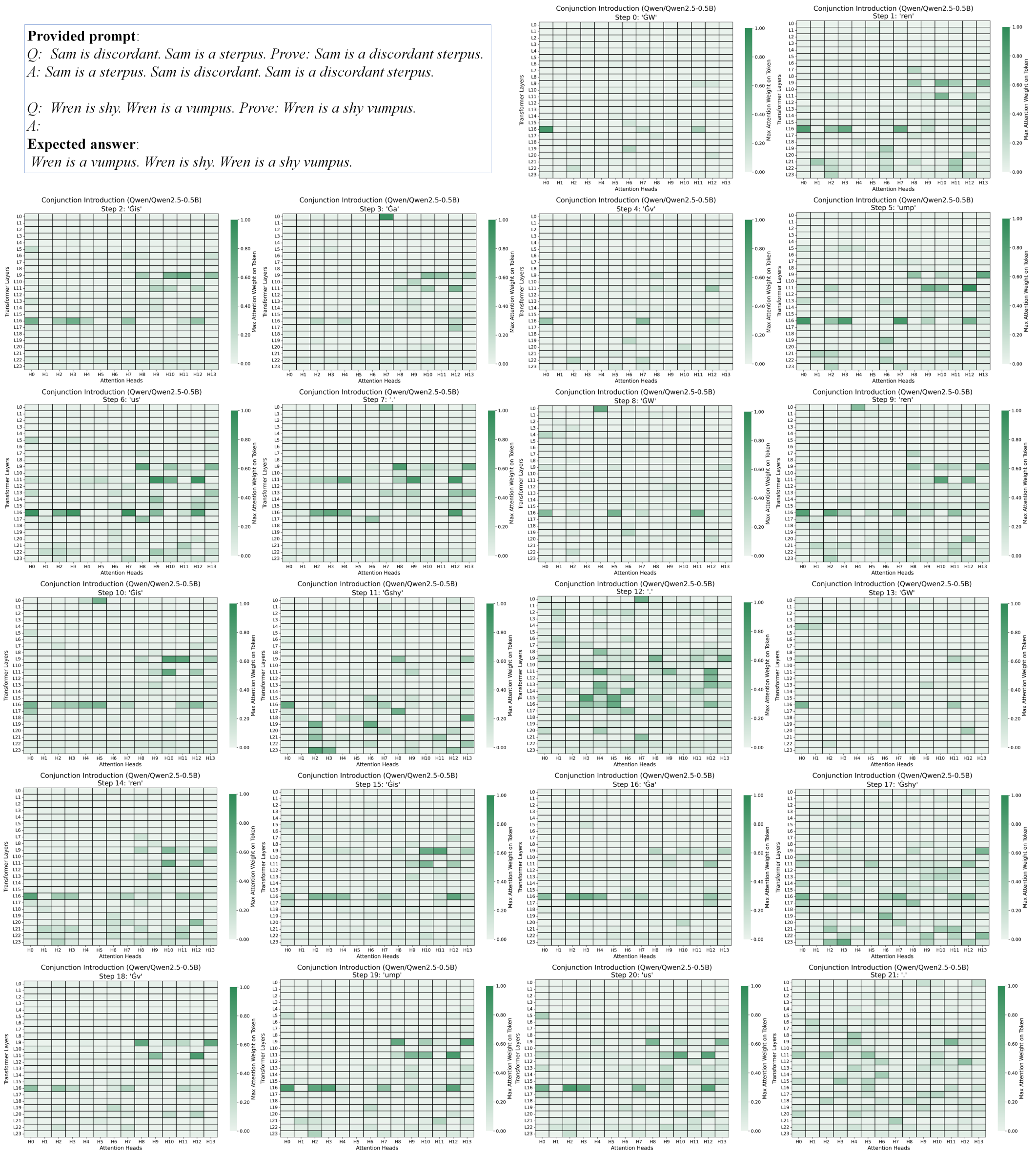
3.2 Results and Discussion on the Zero-shot Commonsense and mathematical Reasoning

We evaluated the zero-shot reasoning ability of 29 decoder-only LMs on the CommonsenseQA dataset [26] and the GSM8K dataset [28]. Each model was tested using three prompting strategies: *Let's think step-by-step*, *Plan-and-Solve*, and *Role-Play*. As shown in Figure 4, models with larger parameter counts and those fine-tuned on instruction data achieve higher accuracy. These results suggest that **the zero-shot reasoning ability of decoder-only transformer-based LMs is depends on model size and instruction-tuning**.

Models trained on code show a strong reasoning ability. Code data is well organized with algorithmic logic and programming flow, which may be useful to improve the reasoning performance of LMs. Reinforcement learning (RL) with a process-based reward model can further guide the LM toward generating logically consistent solutions. Token representations in the initial half layers of LMs remain strongly biased towards the pre-training prior, with the in-context prior taking over in the later half [36], suggesting that deeper layers are more context-aware. Instruction-tuning reduces the amount of prompt engineering and few-shot exemplars required to elicit a useful, accurate response from the fine-tuned model: instruction-tuned LMs, such as those trained on FLAN-style datasets are zero-shot learners [37]. *Let's think step by step* is a zero-shot CoT prompting strategy, whose effectiveness emerges with increasing model size.

Adding additional tasks to the instruction-tuning dataset improves performance even on novel tasks not represented in the original training data. Therein lies the fundamental benefit of instruction tuning: a holistic improvement in the model's ability to follow instructions in general [37]. Instruction finetuning on CoT tasks—both with and without few-shot exemplars—increases a model's ability for CoT reasoning across diverse arithmetical, symbolic reasoning and other logical reasoning tasks in a zero-shot setting. An intuitive understanding of this benefit would be that through being fine-tuned to work through a problem in logical steps rather than leap to an answer that simply seems linguistically coherent, models learn to better produce and apply their own reasoning skills [11]. Furthermore, reasoning capabilities can be transferred to smaller models via knowledge distillation, by fine-tuning a student model on the CoT outputs generated by a larger teacher model [38].

As shown in Table 3, fine-tuning SmolLM2-135M and SmolLM2-135M-Instruct enables near-perfect accuracy on deductive reasoning tasks that require three or fewer reasoning steps when using zero-shot prompting. However, performance declines significantly on tasks requiring seven or more reasoning steps. Notably, the benefits of fine-tuning these models do not generalize to mathematical or commonsense reasoning tasks, indicating that the gains may be domain- or model structure-specific.



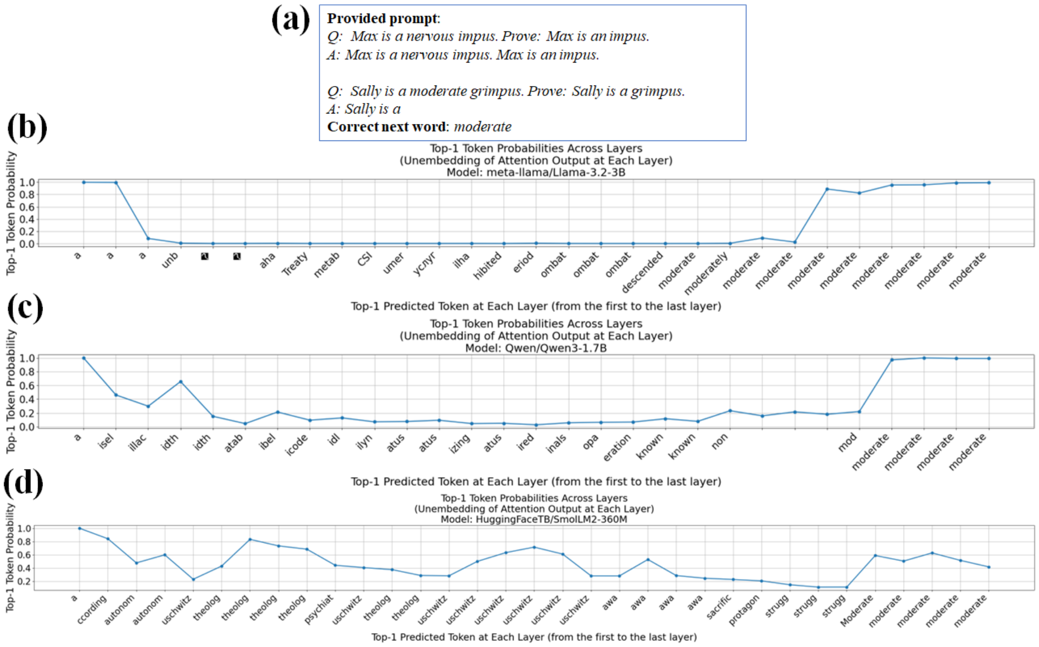
Provided prompt:
Q: Sam is discordant. Sam is a sterpus. Prove: Sam is a discordant sterpus.
A: Sam is a sterpus. Sam is discordant. Sam is a discordant sterpus.

Q: Wren is shy. Wren is a vumpus. Prove: Wren is a shy vumpus.
A:
Expected answer:
Wren is a vumpus. Wren is shy. Wren is a shy vumpus.

Figure 2: Heat maps showing the maximum attention weights corresponding to the correct next token, extracted from the last row of the attention map across all heads and layers. The Qwen2.5-0.5B model is evaluated on a conjunction introduction task with one exemplar. The prompt is incrementally extended by adding one token from the expected answer at each step. These heat maps visualize which attention heads at which layers focus on the correct token. Similar patterns across steps 1, 2, 5, 6, 14, 15, 18, 19, and 20, and steps 3, 7, and 12, suggest that some attention heads may exhibit functional specialization during reasoning.

4 Conclusion

This work systematically investigated the reasoning thresholds of open-source, decoder-only transformer-based language models with fewer than three billion parameters, providing insights across three critical dimensions: ICL, zero-shot reasoning, and the enhancement of zero-shot capabilities in smaller models through fine-tuning.



Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

Figure 3: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for three models: (b) LLaMA-3.2-3B, (c) Qwen3-1.7B, and (d) SmolLM2-360M. The results indicate that the correct token tends to emerge with higher probability at the later layers.

Language Models		Acuracy							
		CommonsenseQA (0-shot)	GSM8K (0-shot)	PrOntoQA-OOD					
				IE (3 steps)	CI (3 steps)	CE (2 steps)	DI (2 steps)	DE (7 steps)	PC (13 steps)
SmolLM2-135M	Base	0.82	1.67	68.00	88.00	95.00	97.00	37.00	5.00
	FT	0.00	0.00	100.00	95.00	100.00	100.00	0.00	1.00
SmolLM2-135M-Instruct	Base	5.00	2.81	77.00	91.00	96.00	98.00	19.00	4.00
	FT	0.00	0.00	98.00	100.00	99.00	99.00	0.00	1.00

Table 1: Comparison of accuracy (%) across various language models on multiple reasoning tasks: The third column shows results on 1,221 multiple-choice questions from the validation set of the CommonsenseQA dataset [26]. The forth column presents accuracy on 1,319 mathematical questions from the GSM8K test set [28]. Columns five through ten show performance on 100 deductive reasoning proofs generated using the PrOntoQA-OOD data generation code [23], using 8-shot prompting on the Base models. The finetuned models are prompted with 0-shot. The proof categories include Implication Elimination (IE), Conjunction Introduction (CI), Conjunction Elimination (CE), Disjunction Introduction (DI), Disjunction Elimination (DE), and Proof by Contradiction (PC). “FT” denotes fine-tuned models and “Base” refers to models that have not been fine-tuned.

Our evaluation of ICL on deductive reasoning tasks revealed that performance is significantly influenced by model scale (parameter count, embedding dimension, feedforward width, number of attention heads, and model depth) and pretraining data characteristics, including its scale, diversity, and the presence of long-range coherence. The ordering of in-context exemplars also plays a role, with ICL accuracy generally increasing up to approximately four shots before plateauing. Notably, standard instruction tuning did not consistently benefit ICL reasoning performance in this regime.

For zero-shot reasoning on commonsense and mathematical tasks, we demonstrated that model size and instruction tuning are primary drivers of performance. Fine-tuning on datasets incorporating instruction and code, leveraging specific prompting strategies such as plan-and-solve and role-play (for instruction-tuned models), and the increased of the number of model parameters were all found

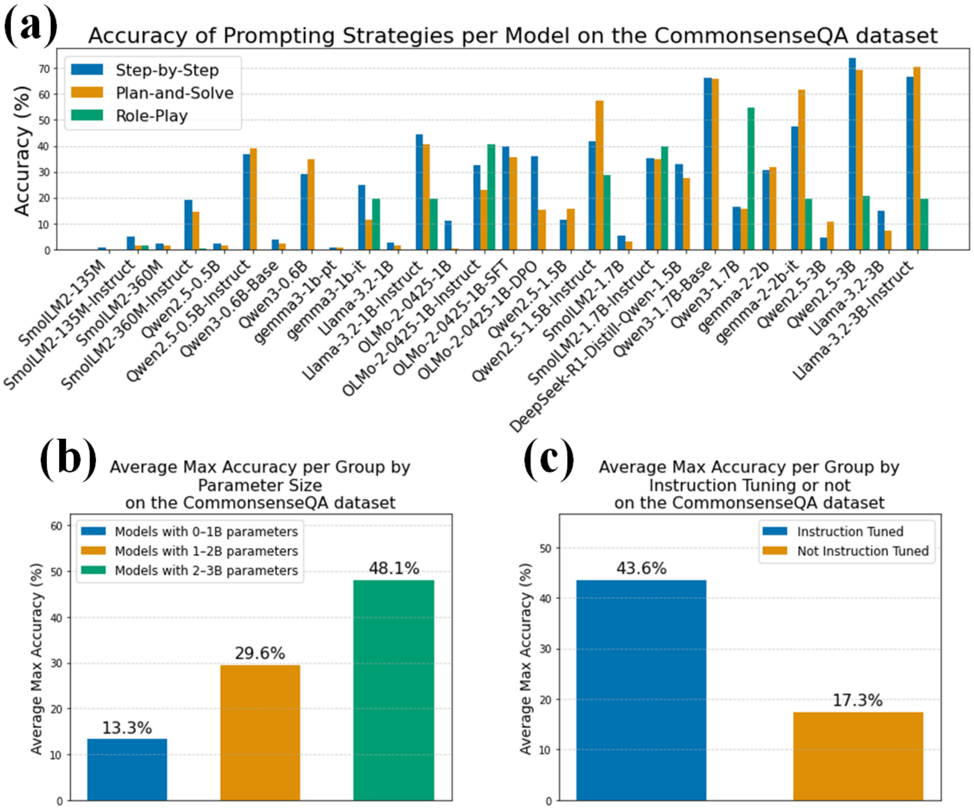


Figure 4: Evaluation results of 29 LMs on the CommonsenseQA dataset [26] using three prompting strategies: *Let's think step-by-step*, *Plan-and-Solve*, and *Role-Play*. Each model is tested for 12 hours or on 1221 questions, whichever comes first. (a) Accuracy of all 29 models under the three prompting strategies. Models not instruction-tuned are excluded from the Role-Play prompting. (b) Averaged maximum accuracy grouped by model size: 0–1B, 1–2B, and 2–3B parameters. For each model, the highest accuracy among the three prompting strategies is used, and the average is computed within each group. The results indicate a general trend of increasing accuracy with model size. (c) Averaged maximum accuracy grouped by instruction tuning. Instruction-tuned models show substantially higher performance compared to those without instruction-tuning.

to contribute positively. These findings underscore that even without exemplars, appropriately scaled and tuned smaller LMs can exhibit significant reasoning capabilities.

Fine-tuning small LMs improved their zero-shot performance on deductive reasoning tasks involving less than three reasoning steps, but the benefit diminished on tasks requiring longer steps. These gains did not generalize to mathematical or commonsense reasoning, suggesting that fine-tuning effects may be specific to task type or model architecture.

To substantiate these findings and offer deeper insights into the internal mechanisms, we conducted attention-based interpretability analyses. Our examination of multi-head attention patterns suggested a correlation with multiple reasoning paths, with certain heads potentially specializing for particular reasoning steps. Attention unembedding analyses further indicated that the correct tokens tend to be represented with higher probability in the outputs of later attention layers, suggesting these deeper layers are more critical in forming the final reasoned output.

In summary, this research delineates key architectural, pretraining, and fine-tuning factors that govern the emergence and enhancement of reasoning in LMs with less than three billion parameters. These insights contribute to a clearer understanding of how to effectively develop and utilize smaller, open-source models for complex reasoning tasks.

Acknowledgments and Disclosure of Funding

The computational work for this project was conducted using resources provided by the Storrs High-Performance Computing (HPC) cluster. We extend our gratitude to the UConn Storrs HPC and its team for their resources and support, which aided in achieving these results.

This research received no external funding.

References

[1] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States, July 2022. Association for Computational Linguistics.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[6] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications, 2022.

[7] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[9] Kevin Christian Wibisono and Yixin Wang. From unstructured data to in-context learning: Exploring what tasks can be learned and when. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[10] Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. Chain of thoughtlessness? an analysis of cot in planning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[11] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022.

[12] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634. Association for Computational Linguistics, 2023.

[13] Gaurav Srivastava, Shuxiang Cao, and Xuan Wang. Towards reasoning ability of small language models, 2025.

[14] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Lewis Tunstall, Agustín Piqueres, Andres Marafioti, Cyril Zakka, Leandro von Werra, and Thomas Wolf. Smollm2 - with great data, comes great performance, 2024.

[15] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom

Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024.

[16] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025.

[17] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[18] 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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin

Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[19] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[20] 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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kam-

badur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao

Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[21] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2025.

[22] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*, 2023.

[23] Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. Testing the general deductive reasoning capacity of large language models using OOD examples. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[24] Tianyi Lorena Yan and Robin Jia. Promote, suppress, iterate: How language models answer one-to-many factual queries, 2025.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[26] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[27] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press, 2017.

[28] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[29] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models, 2019.

[30] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113. Association for Computational Linguistics, 2024.

[31] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. Survey Certification.

[32] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022.

[33] Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 14228–14246. Curran Associates, Inc., 2023.

[34] Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. Why larger language models do in-context learning differently? In *Forty-first International Conference on Machine Learning*, 2024.

[35] 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, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *CoRR*, abs/2209.11895, 2022.

[36] Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *Transactions on Machine Learning Research*, 2024.

[37] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[38] Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781. Association for Computational Linguistics, 2023.

A Language Models

Language Models	Architecture					Type of Training data	Parameters	Release
	Layers	Embeddings	Heads	FFN width	Context Length			
SmolLM2-135M	30	576	9	1536	8192	Web, math, code, **long context**	∼135M	2024
SmolLM2-360M	32	960	15	2560			∼362M	
SmolLM2-1.7B	24	2048	32	8192			∼1.71B	
SmolLM2-135M-Instruct	30	576	9	1536		Finetuned on conversation, instruction following, summarization, rewriting, math, **long context**, knowledge	∼135M	
SmolLM2-360M-Instruct	32	960	15	2560			∼362M	
SmolLM2-1.7B-Instruct	24	2048	32	8192			∼1.71B	
Qwen2.5-0.5B	24	896	14	4864	32768	focuse on math, code, knowledge, **long context**	∼494M	2024
Qwen2.5-1.5B	28	1536	12	8960			∼1.54B	
Qwen2.5-3B	36	2048	16	11008			∼3.09B	
Qwen2.5-0.5B-Instruct	24	896	14	4864		Finetuned on CoT of math, code, **long-response**, instruction following, others	∼494M	
Qwen2.5-1.5B-Instruct	28	1536	12	8960			∼1.54B	
Qwen2.5-3B-Instruct	36	2048	16	11008			∼3.09B	
Qwen3-0.6B-Base	28	1024	16	3072	40960	Multilingual data, codes, **long context**, STEM, reasoning tasks, books	∼494M	2025
Qwen3-1.7B-Base	28	2048	16	6144	131072		∼1.54B	
Qwen3-0.6B	28	1024	16	3072	40960	Finetuned on **long** CoT, reasoning, instruction, others	∼494M	
Qwen3-1.7B	28	2048	16	6144	131072		∼1.54B	
DeepSeek-R1-Distill-Qwen-1.5B	28	1536	12	8960	131072	Distilled from data generated by DeepSeek-R1	∼1.78B	
gemma-2-2b	26	2304	8	9216	8192	Web, code, science	∼2.6B	2024
gemma-2-2b-it	26	2304	8			Finetuned on prompt-response pairs and preference data		
gemma-3-1b-pt	26	1152	4	6912	32768	Multilingual data and others	∼1B	2025
gemma-3-1b-it						Finetuned on instruction following and other data		
Llama-3.2-1B	16	2048	32	8192	131072	Unknown	∼1.24B	2024
Llama-3.2-1B-Instruct	16	2048	32				∼1.24B	
Llama-3.2-3B	28	3072	24				∼3.21B	
Llama-3.2-3B-Instruct	28	3072	24				∼3.21B	
OLMo-2-0425-1B	16	2048	16	8192	4096	Web, academic paper, code, math	∼1.48B	2025
OLMo-2-0425-1B-SFT						Chat, instruction, CoT, math, code		
OLMo-2-0425-1B-DPO						Instruction, synthetic data, part of the SFT data		
OLMo-2-0425-1B-Instruct						After SFT and DPO, RL on math (GSM8K, MATH, others)		

Table 2: Details of all the considered decoder-only transformer-based language models. Models from SmolLM2 [14], Qwen2.5 [17], Qwen3 [18], and Llama-3.2 [20] are categorized as having long-range coherence. Although the pretraining data for Llama-3.2 [20] is not publicly disclosed, we include it in this category because its training using the output logits from Llama-3.1 models [20], which were pre-trained on long-context data.

B Reasoning Ability on Mathematical Tasks

GSM8K [28] contains 1,319 grade-school math word problems requiring multi-step reasoning. We evaluate models under three zero-shot prompting methods: "Let's think step-by-step" [11], Plan-and-Solve [12], and Role-Play [30]. Role-Play prompting [30] is applied only to instruction-tuned models, as its format presumes system-role awareness. Answer extraction follows [12]. For GSM8K, the answer trigger is: "Therefore, the answer (arabic numerals) is most likely ". We extract the first integer after the trigger. If no integer appears, the model is considered not to have answered. As shown in Figure 5, models with larger parameter counts and those fine-tuned on instruction data tend to achieve higher accuracy. These results suggest that **the zero-shot reasoning ability of decoder-only transformer-based language models (LMs) is depends on model size and instruction-tuning**.

C Finetuning on the Reasoning Tasks

To assess how fine-tuning affects In-context learning (ICL) reasoning, we fine-tuned eight small-scale LMs: SmolLM2-135M, SmolLM2-135M-Instruct, SmolLM2-360M, SmolLM2-360M-Instruct, Qwen2.5-0.5B, Qwen2.5-0.5B-Instruct, Qwen3-0.6B-Base, and Qwen3-0.6B on the PrOntoQA-OOD dataset [23]. The training set contains 1800 exemplars (300 per rule × 6 rules), drawn from PrOntoQA-OOD and split into 90% training and 10% validation. We fine-tune with causal language modeling using the Hugging Face Trainer API on the last two layers for each LM. Training uses

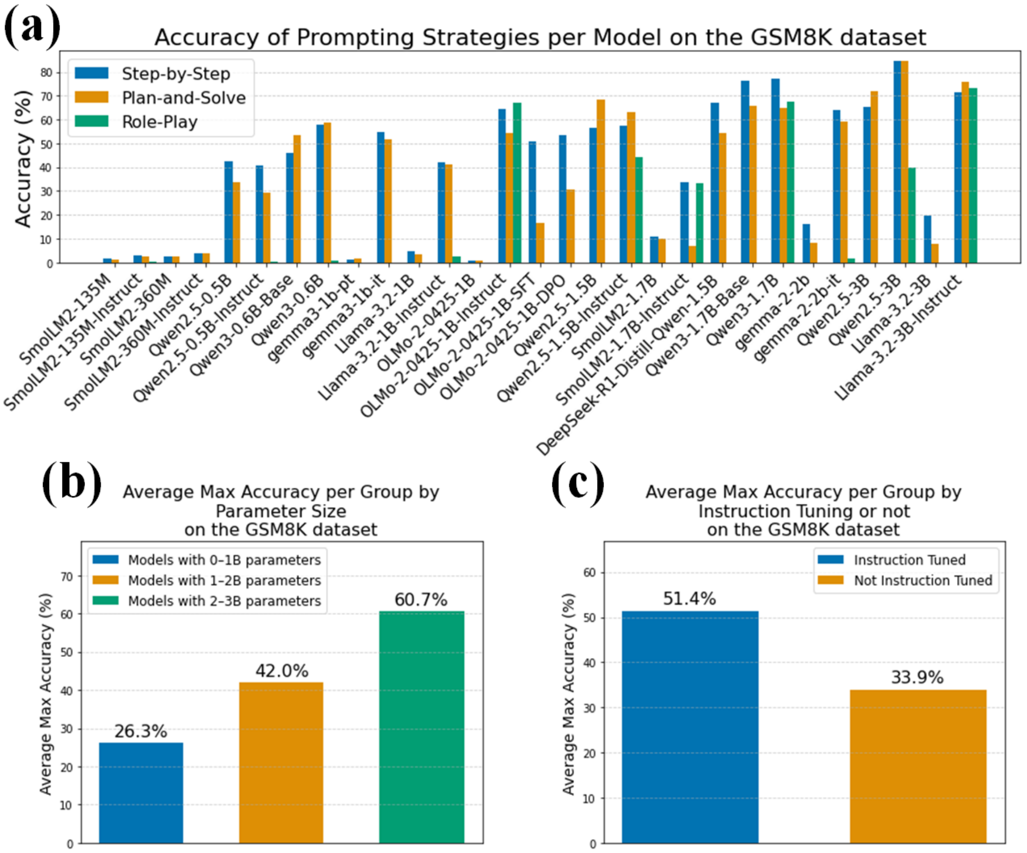
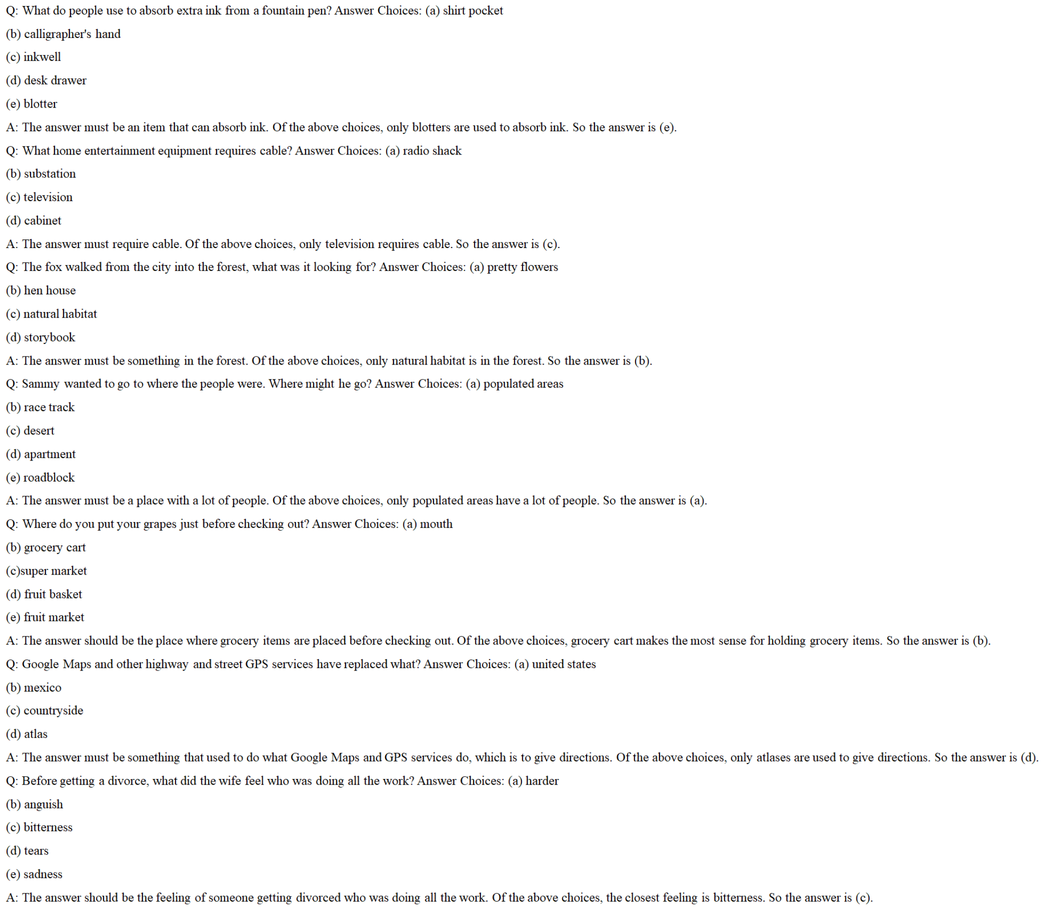


Figure 5: Evaluation results of 29 LMs on the GSM8K dataset [26] using three prompting strategies: *Let's think step-by-step*, *Plan-and-Solve*, and *Role-Play*. Each model is tested for 12 hours or on 1319 questions, whichever comes first. (a) Accuracy of all 29 models under the three prompting strategies. Models not instruction-tuned are excluded from the Role-Play prompting. (b) Averaged maximum accuracy grouped by model size: 0–1B, 1–2B, and 2–3B parameters. For each model, the highest accuracy among the three prompting strategies is used, and the average is computed within each group. The results indicate a general trend of increasing accuracy with model size. (c) Averaged maximum accuracy grouped by instruction tuning. Instruction-tuned models show higher accuracy compared to those without instruction-tuning.

the AdamW optimizer with weight decay 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ [25], for 100 epochs, batch size 8, and learning rate 2×10^{-5}. After training 100 epochs, we select the model checkpoint corresponding to the epoch with the lowest validation loss for evaluation.

To construct the training and validation sets for the CommonsenseQA dataset [26], we used the gemma-2-9b-it model [15] to generate chain-of-thought (CoT) by prompting it with the training set of the CommonsenseQA dataset [26] and the format as shown in Figure 6. We retained only the generated samples where the predicted answer matched the ground-truth answer. In total, we collected 1,000 question–answer pairs, with 900 used for training and 100 for validation. Each sample contains one question and its corresponding CoT answer, and we set the context length to the maximum number of tokens across all 1,000 samples.

The same eight LMs (SmolLM2-135M, SmolLM2-135M-Instruct, SmolLM2-360M, SmolLM2-360M-Instruct, Qwen2.5-0.5B, Qwen2.5-0.5B-Instruct, Qwen3-0.6B-Base, and Qwen3-0.6B) are fine-tuned on the 1,319-sample GSM8K test set [28]. Each training sample consists of a question and a final answer from [28], with all "\ n#### " markers replaced by "The answer is ".



Q: What do people use to absorb extra ink from a fountain pen? Answer Choices: (a) shirt pocket

(b) calligrapher's hand

(c) inkwell

(d) desk drawer

(e) blotter

A: The answer must be an item that can absorb ink. Of the above choices, only blotters are used to absorb ink. So the answer is (e).

Q: What home entertainment equipment requires cable? Answer Choices: (a) radio shack

(b) substation

(c) television

(d) cabinet

A: The answer must require cable. Of the above choices, only television requires cable. So the answer is (c).

Q: The fox walked from the city into the forest, what was it looking for? Answer Choices: (a) pretty flowers

(b) hen house

(c) natural habitat

(d) storybook

A: The answer must be something in the forest. Of the above choices, only natural habitat is in the forest. So the answer is (b).

Q: Sammy wanted to go to where the people were. Where might he go? Answer Choices: (a) populated areas

(b) race track

(c) desert

(d) apartment

(e) roadblock

A: The answer must be a place with a lot of people. Of the above choices, only populated areas have a lot of people. So the answer is (a).

Q: Where do you put your grapes just before checking out? Answer Choices: (a) mouth

(b) grocery cart

(c)super market

(d) fruit basket

(e) fruit market

A: The answer should be the place where grocery items are placed before checking out. Of the above choices, grocery cart makes the most sense for holding grocery items. So the answer is (b).

Q: Google Maps and other highway and street GPS services have replaced what? Answer Choices: (a) united states

(b) mexico

(c) countryside

(d) atlas

A: The answer must be something that used to do what Google Maps and GPS services do, which is to give directions. Of the above choices, only atlases are used to give directions. So the answer is (d).

Q: Before getting a divorce, what did the wife feel who was doing all the work? Answer Choices: (a) harder

(b) anguish

(c) bitterness

(d) tears

(e) sadness

A: The answer should be the feeling of someone getting divorced who was doing all the work. Of the above choices, the closest feeling is bitterness. So the answer is (c).

Figure 6: 7 examples of the format for finetuning on the CommonsenseQA dataset [26]. The 7 examples are taken from [4] and are used to prompt the gemma-2-9b-it model [15] for generating CoT data for finetuning.

For the commonsense and mathematical reasoning tasks, All models are trained for 100 epochs using AdamW with a learning rate of 2×10^{-5}, weight decay of 0.01, batch size of 1, and a linear learning rate schedule Ȧfter training 100 epochs, we select the model checkpoint corresponding to the epoch with the lowest validation loss for evaluation.

As shown in Table 3, fine-tuning SmolLM2-135M, SmolLM2-135M-Instruct, SmolLM2-360M, SmolLM2-360M-Instruct, Qwen2.5-0.5B, Qwen2.5-0.5B-Instruct, Qwen3-0.6B-Base, and Qwen3-0.6B enables near-perfect accuracy on deductive reasoning tasks that require three or fewer reasoning steps when using zero-shot prompting. However, performance declines significantly on tasks requiring seven or more reasoning steps, except for the Qwen2.5-0.5B model on the disjunction elimination rule. Notably, the benefits of fine-tuning these models do not generalize to mathematical or commonsense reasoning tasks, indicating that the gains may be domain- or model structure-specific.

D Attention Unembedding

To determine which tokens the head writes to the residual stream, we apply attention unembedding to the normalized output of each attention layer as in [24]. Let H denote the attention output at a given layer. We project this output to the vocabulary space using

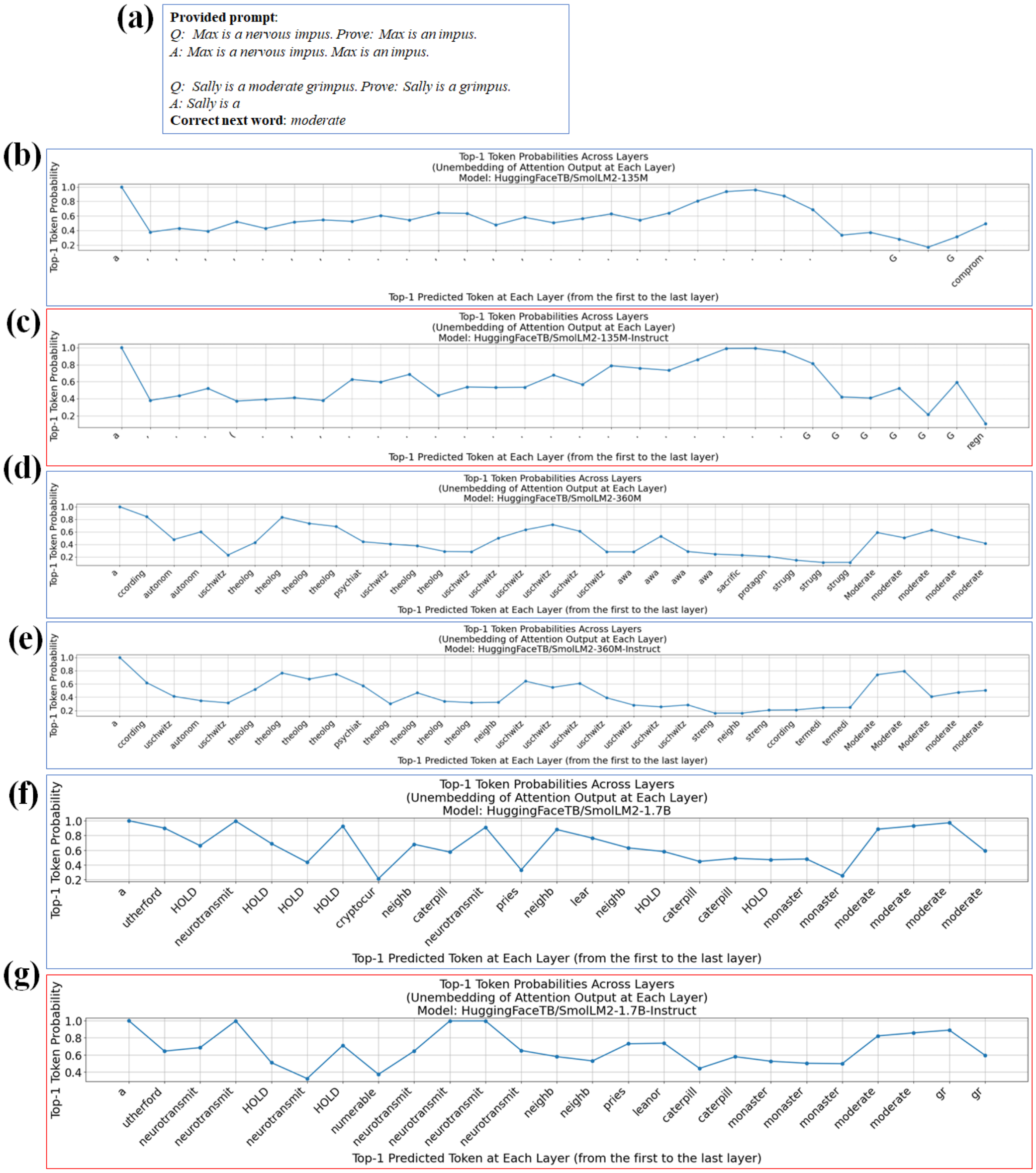
$$Y = \text{LayerNorm}(H)W^{\text{last}}, \tag{5}$$

where $W^{\text{last}} \in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}$ is the unembedding matrix and $|\mathcal{V}|$ is the vocabulary size.

Language Models		Acuracy (%)							
		CommonsenseQA (0-shot)	GSM8K (0-shot)	PrOntoQA-OOD (8-shot)					
				IE (3 steps)	CI (3 steps)	CE (2 steps)	DI (2 steps)	DE (7 steps)	PC (13 steps)
SmolLM2-135M	Base	0.82	1.67	68.00	88.00	95.00	97.00	37.00	5.00
	FT	0.00	0.00	100.00	95.00	100.00	100.00	0.00	1.00
SmolLM2-135M-Instruct	Base	5.00	2.81	77.00	91.00	96.00	98.00	19.00	4.00
	FT	0.00	0.00	98.00	100.00	99.00	99.00	0.00	1.00
SmolLM2-360M	Base	2.29	2.73	98.00	100.00	100.00	100.00	72.00	5.00
	FT	0.00	0.00	100.00	100.00	99.00	98.00	0.00	1.00
SmolLM2-360M-Instruct	Base	19.08	3.94	100.00	100.00	100.00	100.00	53.00	8.00
	FT	0.00	0.00	98.00	99.00	96.00	100.00	0.00	1.00
Qwen2.5-0.5B	Base	2.21	42.46	98.00	100.00	100.00	100.00	95.00	12.00
	FT	0.00	0.00	100.00	100.00	100.00	100.00	0.00	21.00
Qwen2.5-0.5B-Instruct	Base	38.90	40.64	93.00	100.00	100.00	100.00	94.00	26.00
	FT	0.00	0.00	100.00	99.00	100.00	99.00	0.00	19.00
Qwen3-0.6B-Base	Base	3.69	53.45	100.00	100.00	100.00	100.00	100.00	100.00
	FT	0.00	0.00	100.00	99.00	100.00	100.00	0.00	29.00
Qwen3-0.6B	Base	34.97	58.83	100.00	100.00	100.00	100.00	100.00	91.00
	FT	0.00	0.00	97.00	97.00	98.00	100.00	0.00	8.00

Table 3: Comparison of accuracy (%) across various language models on multiple reasoning tasks: The third column shows results on 1,221 multiple-choice questions from the validation set of the CommonsenseQA dataset [26]. The forth column presents accuracy on 1,319 mathematical questions from the GSM8K test set [28]. Columns five through ten show performance on 100 deductive reasoning proofs generated using the PrOntoQA-OOD data generation code [23], using 8-shot prompting on the Base models. The finetuned models are prompted with 0-shot. The proof categories include Implication Elimination (IE), Conjunction Introduction (CI), Conjunction Elimination (CE), Disjunction Introduction (DI), Disjunction Elimination (DE), and Proof by Contradiction (PC). "FT" denotes fine-tuned models and "Base" refers to models that have not been fine-tuned.

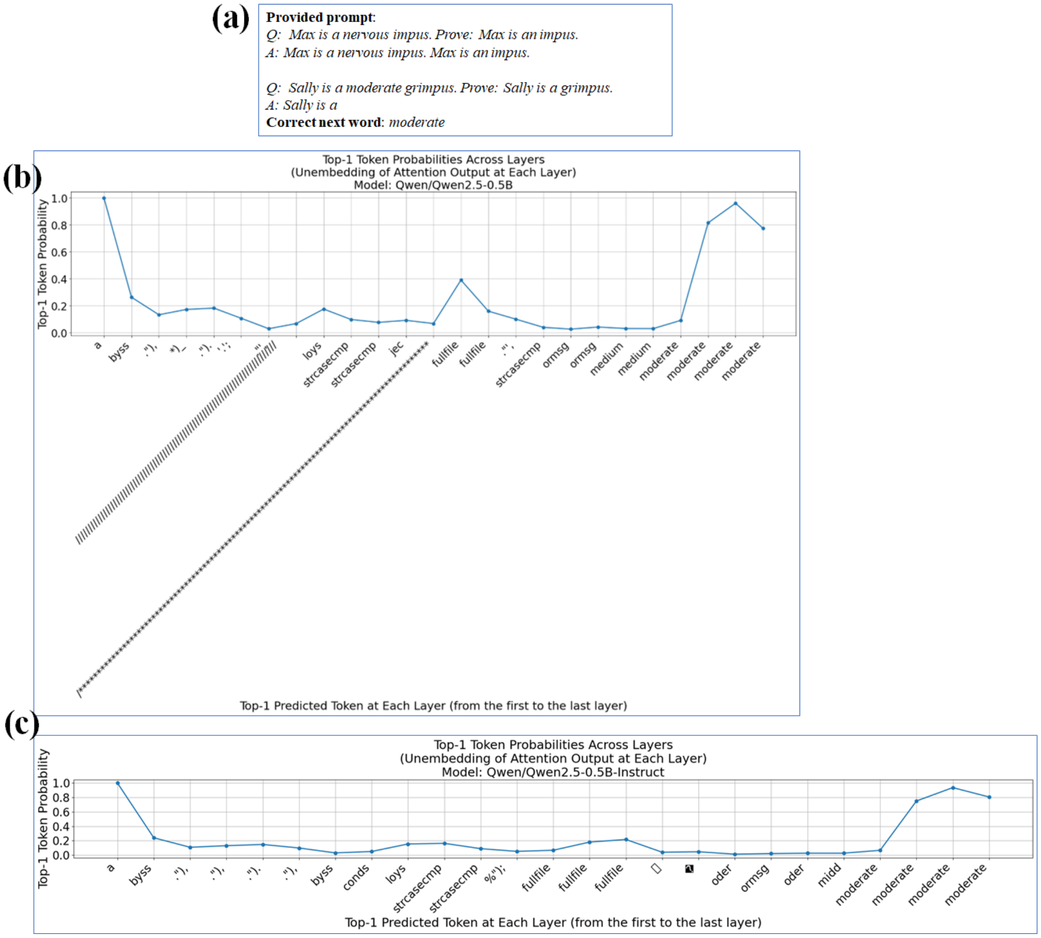
To determine which token attention heads contributes to in the residual stream, we perform attention unembedding on the output of each attention layer. Specifically, for each layer, we project the attention output into the vocabulary space and identify the token with the highest probability. We apply this procedure to a conjunction elimination task with one exemplar, where the correct next token is "moderate," as shown from Figure 7 to 13. The analysis is conducted on three different language models: LLaMA-3.2-3B, Qwen3-1.7B, and SmolLM2-360M, all of which correctly predict the next token using greedy decoding.



Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

Figure 7: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for six models: (b) SmolLM2-135M, (c) SmolLM2-135M-Instruct, (d) SmolLM2-360M, (e) SmolLM2-360M-Instruct, (f) SmolLM2-1.7B, and (g) SmolLM2-1.7B-Instruct. The results indicate that the correct token tends to emerge with higher probability at the later layers. Plots with a red border indicate that the corresponding model produced an incorrect prediction, while plots with a blue border indicate a correct prediction.



Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

Figure 8: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for two models: (b) Qwen2-0.5B, and (c) Qwen2-0.5B-Instruct. The results indicate that the correct token tends to emerge with higher probability at the later layers.

(a)

Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

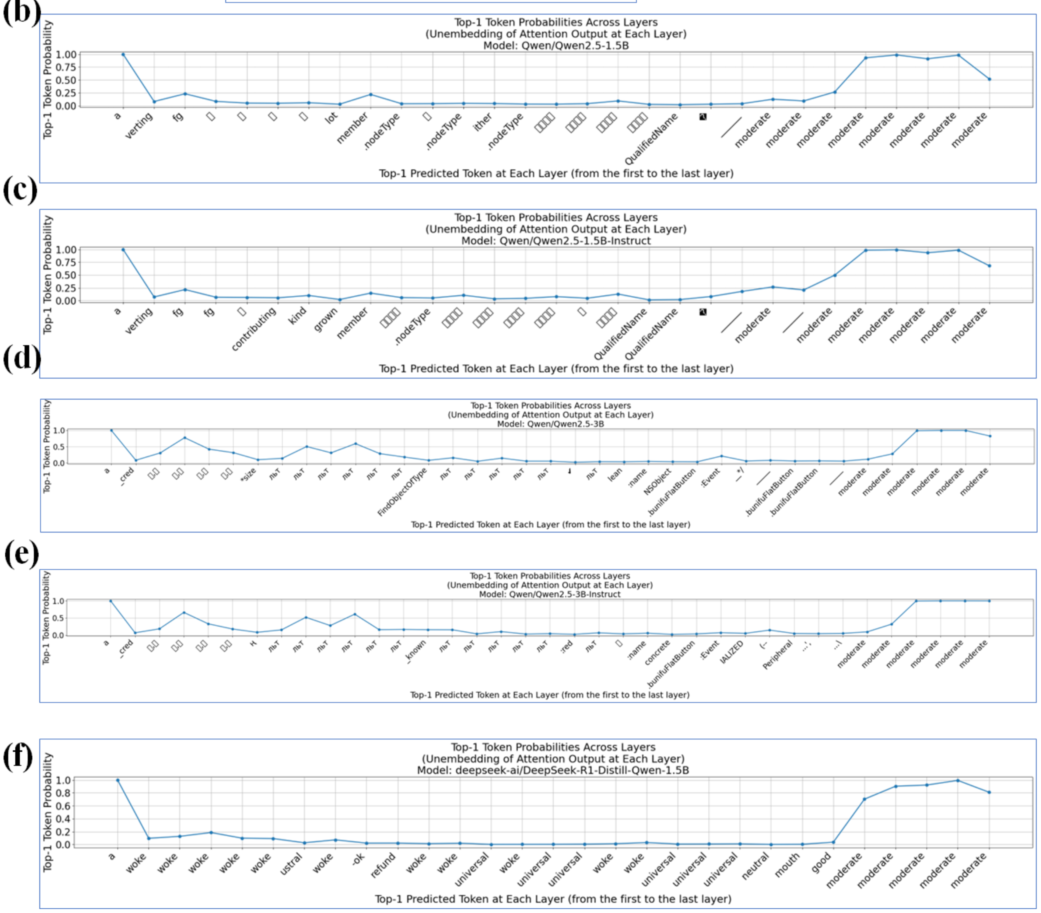
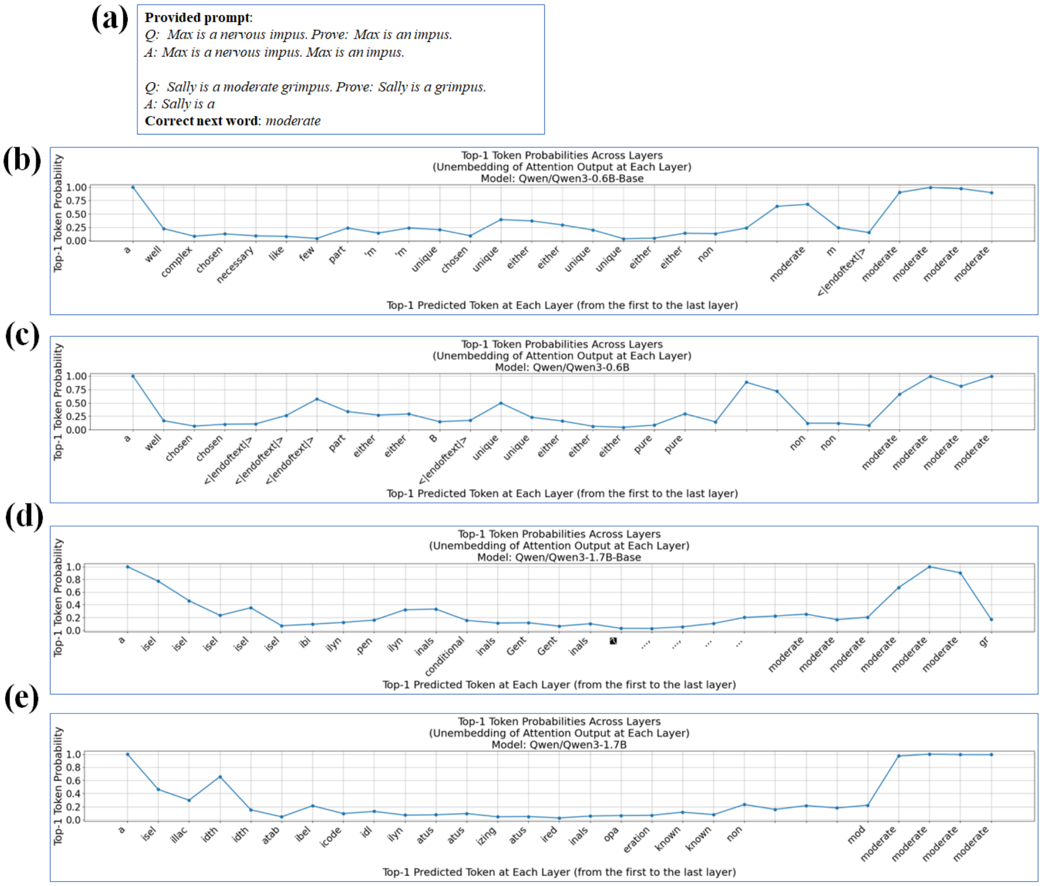


Figure 9: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for five models: (b) Qwen2-0.5B, (c) Qwen2-0.5B-Instruct, (d) Qwen2-1.5B, (e) Qwen2-1.5B-Instruct, and (f) DeepSeek-R1-Distill-Qwen-1.5B. The results indicate that the correct token tends to emerge with higher probability at the later layers.

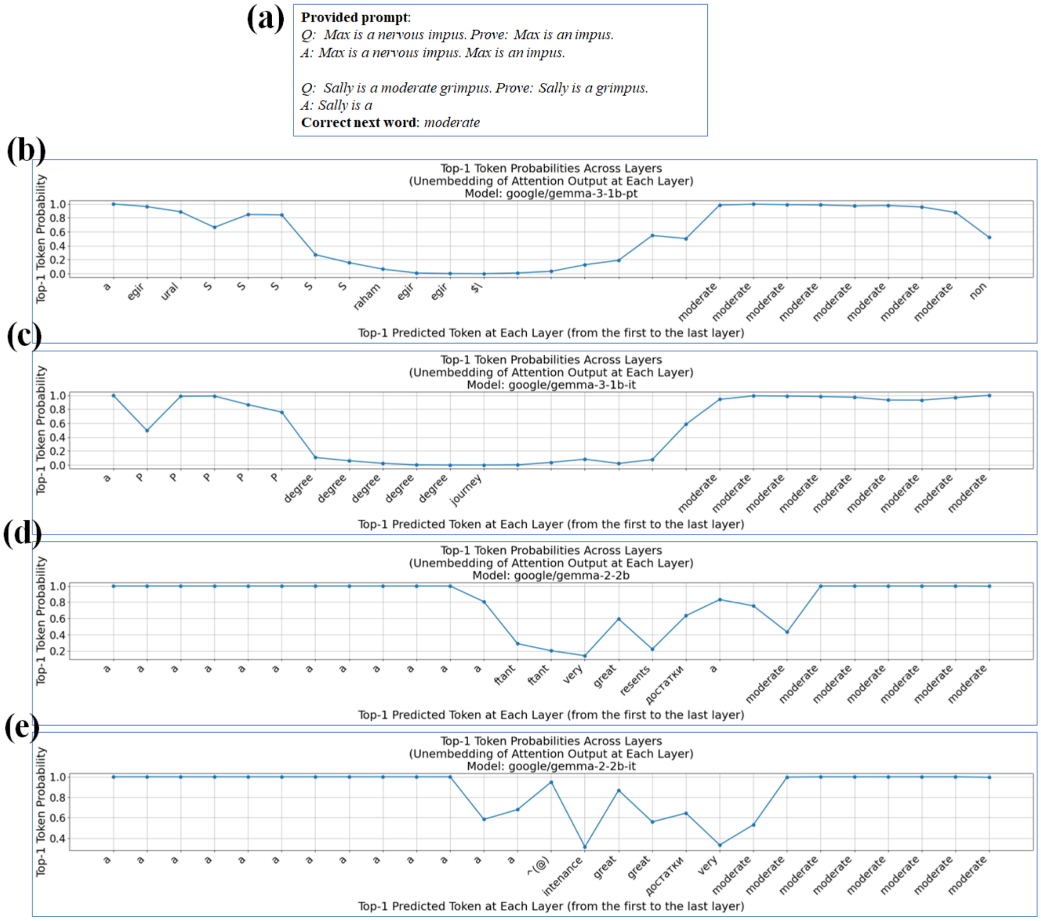


(a)

Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

Figure 10: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for four models: (b) Qwen3-0.6B-Base, (c) Qwen3-0.6B, (d) Qwen3-1.7B-Base, and (e) Qwen3-1.7B. The results indicate that the correct token tends to emerge with higher probability at the later layers.



Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

Figure 11: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for four models: (b) gemma-3-1b-pt, (c) gemma-3-1b-it, (d) gemma-2-2b, and (e) gemma-2-2b-it. The results indicate that the correct token tends to emerge with higher probability at the later layers.

(a)

Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

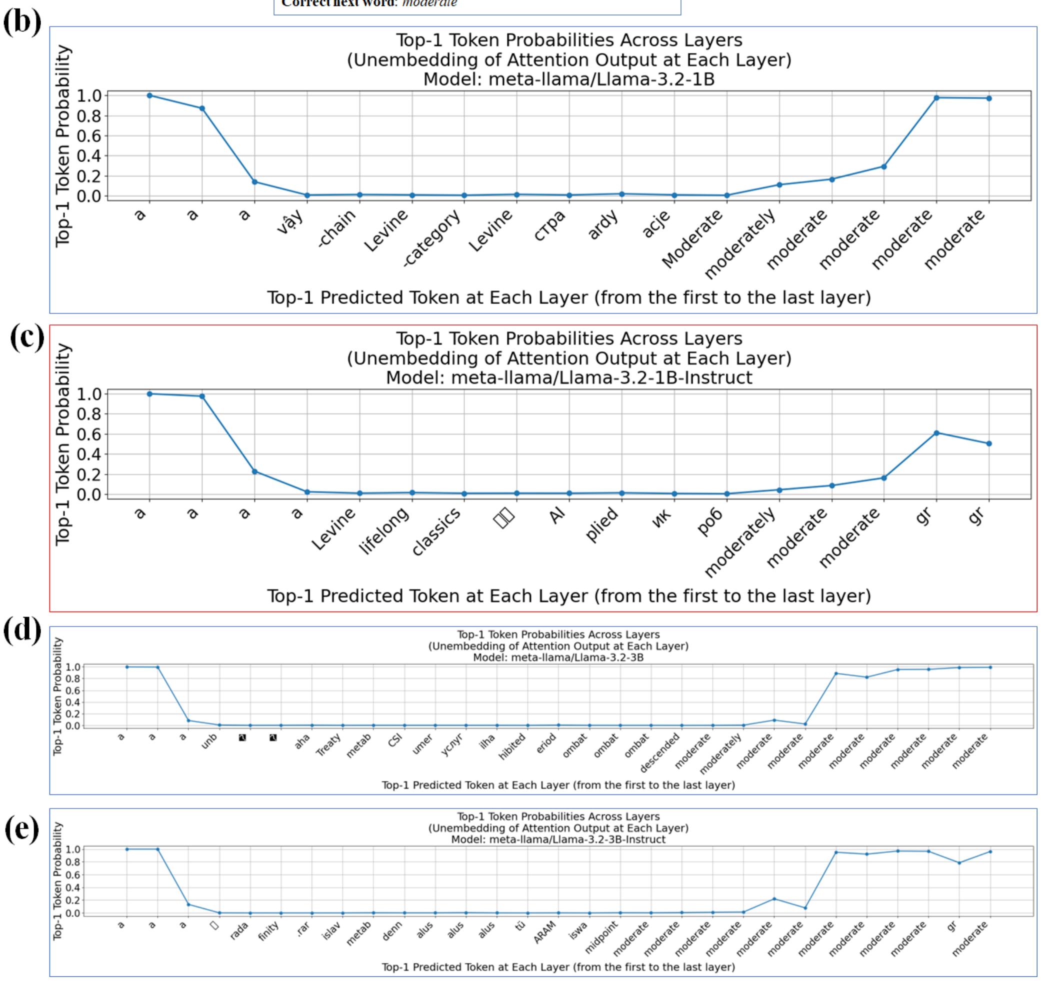
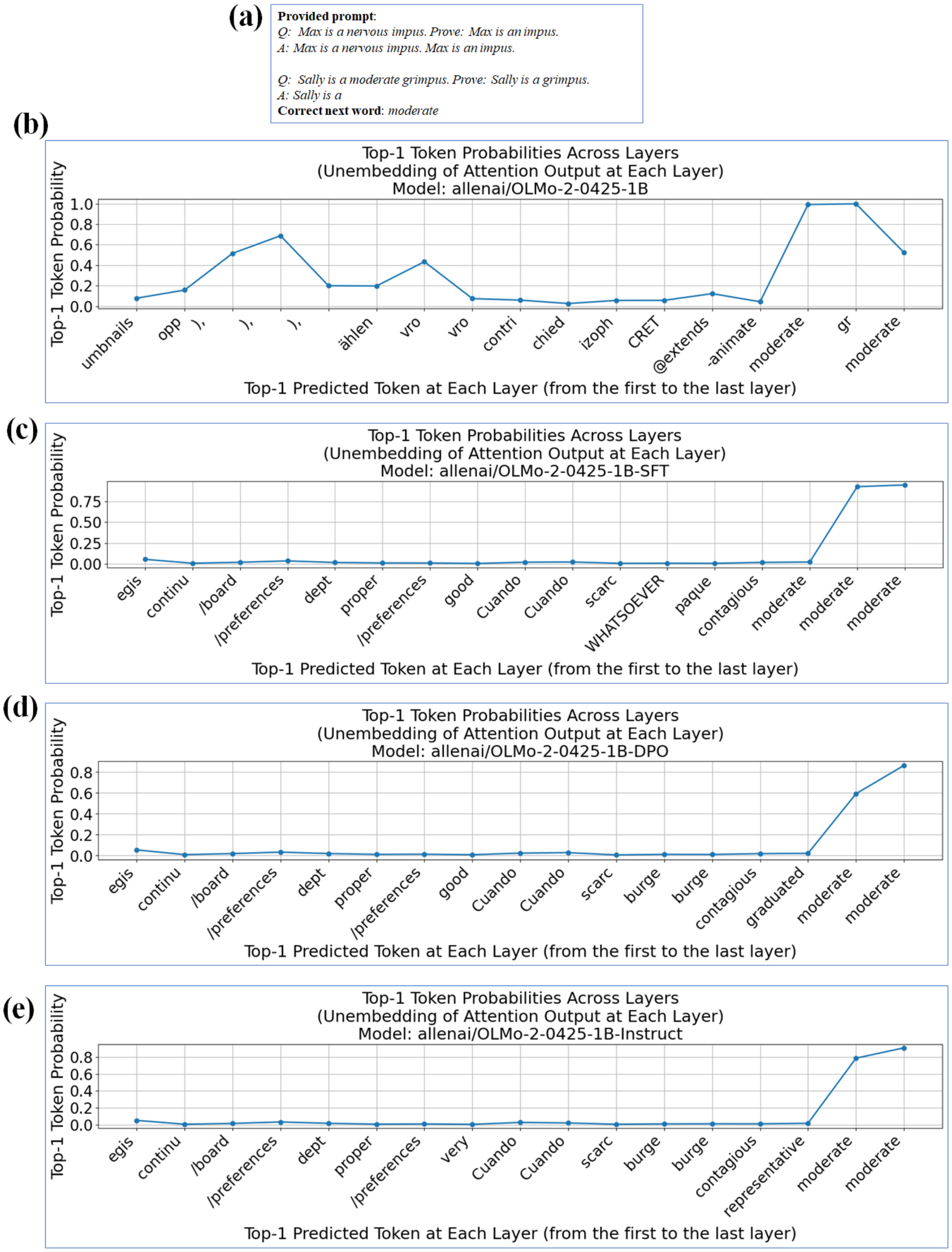


Figure 12: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for four models: (b) LLaMA-3.2-1B, (c) LLaMA-3.2-1B-Instruct, (d) LLaMA-3.2-3B, and (e) LLaMA-3.2-3B-Instruct. The results indicate that the correct token tends to emerge with higher probability at the later layers. Plots with a red border indicate that the corresponding model produced an incorrect prediction, while plots with a blue border indicate a correct prediction.



Provided prompt:
Q: Max is a nervous impus. Prove: Max is an impus.
A: Max is a nervous impus. Max is an impus.

Q: Sally is a moderate grimpus. Prove: Sally is a grimpus.
A: Sally is a
Correct next word: *moderate*

Figure 13: Line plots showing the probability of the correct next token obtained by unembedding the output of the attention layer at each layer. Given the conjunction elimination input with the correct next token "moderate" (shown in subfigure (a)), we present the attention unembedding results for four models: (b) OLMo-2-0425-1B, (c) OLMo-2-0425-1B-SFT, (d) OLMo-2-0425-1B-DPO, and (e) OLMo-2-0425-1B-Instruct. The results indicate that the correct token tends to emerge with higher probability at the later layers.