Structured Pruning for Diverse Best-of-N Reasoning Optimization

Hieu Trung Nguyen, Bao Nguyen, Viet Anh Nguyen
The Chinese University of Hong Kong
{thnguyen, nbnguyen, nguyen}@se.cuhk.edu.hk

Abstract

Model pruning in transformer-based language models, traditionally viewed as a means of achieving computational savings, can enhance the model's reasoning capabilities. In this work, we uncover a surprising phenomenon: the selective pruning of certain attention heads leads to improvements in reasoning performance, particularly on challenging tasks. Motivated by this observation, we propose SPRINT, a novel contrastive learning framework that dynamically selects the optimal head and layer to prune during inference. By aligning question embeddings with head embeddings, SPRINT identifies those pruned-head configurations that result in more accurate reasoning. Extensive experiments demonstrate that our method significantly outperforms traditional best-of-N and random head selection strategies on the MATH500 and GSM8K datasets.

1 Introduction

Language models (LMs) have recently made remarkable progress in understanding and reasoning with human language, attracting considerable research attention (Brown et al., 2020; Hoffmann et al., 2022; Chowdhery et al., 2022; Rae et al., 2021; Raffel et al., 2019). This unprecedented evolution opened up a wide range of practical applications, including content generation (Li et al., 2024a; Liu et al., 2023b; Achiam et al., 2023), virtual assistance (Sezgin, 2024; Doe et al., 2023; Garcia et al., 2023), population simulation (Bui et al., 2025), and problem solving (Ahn et al., 2024; Imani et al., 2023; Lewkowycz et al., 2022).

Despite these advances, the reliability and correctness of LM-generated answers remain a significant concern. Analyses of their output frequently reveal faulty reasoning and factual inaccuracies (Ji et al., 2023; Xu et al., 2024; Nguyen et al., 2025a; Jiang et al., 2025; Li et al., 2024b). This issue is particularly pronounced in tasks that require advanced reasoning, such as automated theorem proving (Wu et al., 2022), mathematical problem solving (Trinh et al., 2024), or heuristic discovery (Romera-Paredes et al., 2024). LLMs often struggle to produce accurate responses in these scenarios in a single pass.

To address this challenge, iterative generation strategies have been employed to refine and select the most appropriate response, often by combining language generation with aggregation or search techniques such as best-of-N sampling (Stiennon et al., 2020) or Monte Carlo Tree Search (Xie et al., 2024). Throughout this paper, we refer to these iterative approaches as reasoning methods.

Current reasoning methods typically leverage the diversity induced by the LLM's decoding process through parameters such as temperature, top-k, and top-p to generate multiple candidate answers (Stiennon et al., 2020; Xie et al., 2024). Although this technique introduces textual diversity and paraphrasing, it does not capture fundamentally different viewpoints, since all candidates are produced by the same underlying model with similar reasoning capabilities. One potential remedy is to employ a variety of LMs to generate answers; however, this approach is often computationally expensive and memory inefficient.

At the same time, recent research on model pruning has shown that comparable performance can be achieved by selectively disabling certain heads in the transformer architecture (Ma et al., 2023; Wang et al., 2019). Removing an attention head does not necessarily weaken the overall ability of the model because a head may introduce noise into the decision-making process, leading to *in*correct answers. In fact, our empirical observations suggest that pruning certain heads can *improve* the reasoning methods' performance on specific categories of questions; Section 3.2 discusses this phenomenon in more detail. This finding motivates our approach: we aim to construct a pool of pruned models from

Findings of the Association for Computational Linguistics: ACL 2025, pages 23911–23922
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

one base model with diverse reasoning capabilities without incurring additional computational or memory overhead.

Contributions. We begin by analyzing and providing new insights into the surprising phenomenon that pruning attention heads can improve the performance of the language model on the MATH benchmark. Motivated by these insights, we introduce the Structured PRunIng for diverSe reasoNing opTimization (SPRINT) framework. SPRINT first learns a set of embeddings for each layer-attention head pair using contrastive learning, guided by a novel Diversity-Promoted Contrastive Loss. This loss function encourages the question embedding to move closer to the embeddings of heads whose pruning leads to correct answers for that question. Finally, we leverage SPRINT to enhance best-of-N sampling by pruning the top N nearest layer-model configurations, determined by the distance between the learned embeddings and the question embedding.

Experiments on the MATH500 and GSM8K datasets show that our method outperforms traditional best-of-N sampling approaches, achieving higher accuracy without introducing inference-time overhead.

Our paper unfolds as follows: Section 2 discusses related work on reasoning in LMs. Section 3 investigates the impact of pruning individual attention heads in transformer models, revealing that certain heads can be removed without performance loss and, in some cases, even improve accuracy. Section 4 delineates our SPRINT framework for matching heads for pruning with input questions, and Section 5 presents the extensive numerical results of the mathematical reasoning task.

2 Related Works

A straightforward decoding process may struggle to produce accurate solutions for complex reasoning tasks. To address this, Self-Consistency (SC) improves reliability by generating multiple outputs and selecting the final response by majority voting (Wang et al., 2022). A similar approach, best-of-N sampling, employs a reward model or function to select the response with the highest reward (Stiennon et al., 2020). While both techniques enhance output quality, they also increase computational cost proportional to the number of samples.

To explore reasoning pathways more effectively, tree-search-based methods have been introduced, such as Tree-of-Thought (Yao et al., 2024), Monte Carlo Tree Search (MCTS) (Feng et al., 2023; Zhang et al., 2024; Guan et al., 2025), and Forest-of-Thought (Bi et al., 2024). Damani et al. (2024) highlights that tree-based search strategies are more effective in discovering correct solutions than parallel sampling, particularly for complex problems.

Beyond reasoning strategies, scaling inference-time computation also plays a crucial role in improving output quality. Beeching et al. (2024) shows that the accuracy on the MATH-500 benchmark improves as the test-time computation (that is, the number of generations per problem) increases for methods such as best-of-N, beam search, and diverse verifier tree search (DVTS). Similarly, Guan et al. (2025) achieved an average accuracy of 53.3% on the USA Math Olympiad (AIME) by conducting extensive MCTS rollouts.

Our work is related to recent research on interventions in the internal state of language models (Wu et al., 2024; Li et al., 2023; Nguyen et al., 2025b; Liu et al., 2023a; Nguyen et al., 2025c). Although prior approaches often focus on learning intervention vectors to add to model activations, instead we investigate the effect of excluding specific components of the internal state of the model.

3 Does Pruning an Attention Head Improve LM's Performance?

We consider a transformer-based model, where each layer contains H attention heads. In each layer ℓ, the outputs of the H attention heads are denoted as $\{a_h^\ell\}_{h=1}^H$, where each $a_h^\ell \in \mathbb{R}^d$. The concatenated vector, $a^\ell = (a_1^\ell, \ldots, a_H^\ell) \in \mathbb{R}^D$ with $D = H \times d$, is then passed to a linear layer to produce the output of a multi-head attention block.

Attention head pruning. In our study, we define **head pruning** as zeroing out the output vector a_h^ℓ of a specific attention head before the concatenation step, as illustrated in Figure 1. In modern decoder-only architectures (), this is equivalent to pruning the input of $\mathtt{o}_{\mathtt{proj}}$ layer.

In this section, we detail our experimental procedure for attention head pruning and present a surprising empirical phenomenon that motivates our subsequent work.

3.1 Experimental Procedure

To systematically examine the effect of individual head pruning, we follow the procedure below:

1. **Layer and Head Selection:** We select $L = 4$

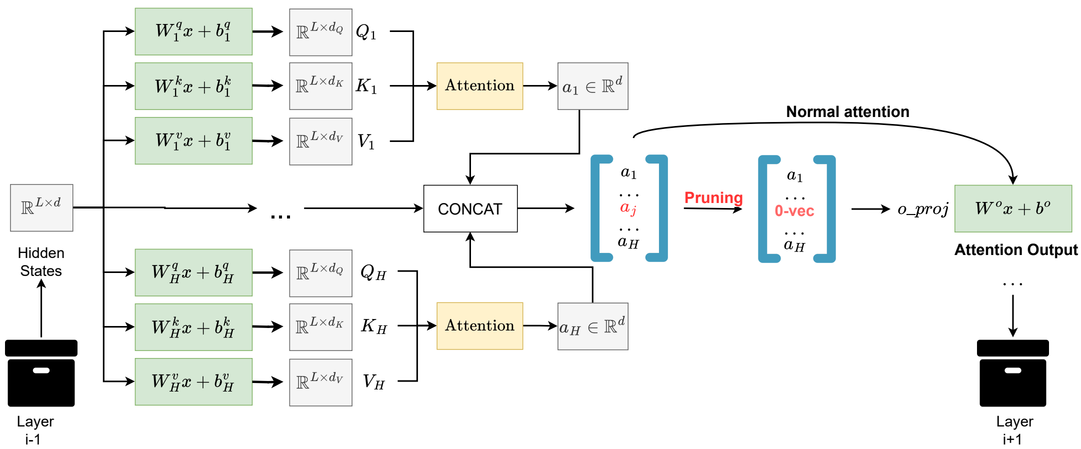

Figure 1: Illustration of our attention head pruning technique. Most of the plot depicts the standard multi-head attention mechanism in transformer-based models. Our approach differs by pruning a specific attention head a_j—that is, we zero out its output vector before concatenating the head outputs and passing the result to the $\mathtt{o_{proj}}$ layer.

layers from the model, yielding a total of LH attention heads. For each head, we create a variant of the base model with that specific head pruned, resulting in LH model variants in addition to the base model. We denote this model pool as $\mathcal{M} = \{\mathcal{M}_{\text{base}}\} \cup \{\mathcal{M}_h^\ell\}$.

2. **Measuring the Impact:** For each model $m \in \mathcal{M}$, we generate solutions on the MATH500 dataset, which consists of $n = 500$ samples. The predictions are represented by a binary accuracy vector $z \in \{0,1\}^n$, where $z_i = 1$ if sample i is solved correctly and 0 otherwise. Evaluating all models yields a matrix $Z \in \{0,1\}^{n \times 4H}$ of accuracy values.

In this experiment, we study two models specialized for mathematical reasoning, `Qwen2.5-Math-1.5B-Instruct` and `Qwen2.5-Math-7B-Instruct`, as well as two general-purpose models, `Qwen2.5-7B-Instruct`, and `Meta-Llama-3-8B-Instruct`. Our experiments involve evaluating model performance with and without head pruning on 500 samples from the MATH500 dataset (Hendrycks et al., 2021).

3.2 Empirical Observations

Our empirical results, shown in Tables 2, 3, 4, and 1, reveal several important and sometimes surprising findings about attention head pruning. Contrary to the common belief that all attention heads are essential for optimal model performance, we find that pruning individual heads often does not reduce accuracy, and in many cases, actually increases it. The impact of pruning depends strongly on both the specific head and its layer: while some heads are crucial (with pruning causing sharp drops in accuracy), many others are unnecessary, and removing them can leave performance unchanged or even lead to notable improvements. For instance, in the 'Counting & Probability' category, pruning head 0 in layer 15 of `Qwen2.5-Math-1.5B-Instruct` increased accuracy by 16% over the unpruned baseline.

We further illustrate this trend in Figure 2, which displays violin plots of additive performance gains (the difference between the best pruned head accuracy and the baseline accuracy) across all subcategories for each model. These plots show that head pruning consistently produces positive gains for all evaluated models. The specialized mathematical models (`Qwen2.5-Math-1.5B-Instruct` and `Qwen2.5-Math-7B-Instruct`) exhibit higher median gains and more frequent large improvements, particularly in domains such as 'Counting & Probability' and 'Geometry'. In contrast, the general-purpose models (`Qwen2.5-7B-Instruct` and `Meta-Llama-3-8B-Instruct`) show more modest but still positive and consistent benefits from pruning.

Overall, these patterns suggest that some attention heads contribute redundant or even harmful information, and that pruning can act as a useful form of regularization. Our findings highlight the significant redundancy present in multi-head attention and indicate promising directions for simplifying models and improving efficiency through targeted head pruning.

4 Methodology

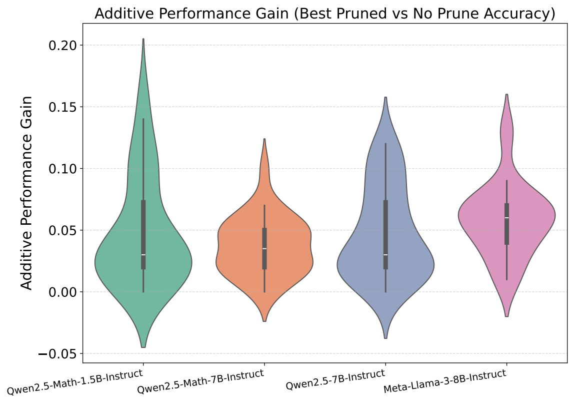

Figure 2: Distribution of additive performance gains (best pruned accuracy minus no-prune accuracy) for Qwen2.5-Math-1.5B-Instruct, Qwen2.5-Math-7B-Instruct, Qwen2.5-7B-Instruct, and Meta-Llama-3-8B-Instruct. Each violin illustrates the improvement across all subject-layer pairs.

Motivated by these counterintuitive findings, we propose a framework called Structured Pruning for Diverse Reasoning Optimization (SPRINT) to leverage selective head pruning to improve reasoning performance. At the core of SPRINT is a contrastive learning approach that aligns question embeddings with attention head embeddings to identify the heads most beneficial to prune. Initially, every head is associated with a learnable vector from a set $\mathcal{V} = \{v_j\}_{j=1}^{LH}$, where each $v_j \in \mathbb{R}^p$ is randomly initialized. A sentence embedding model, denoted by ϕ, processes each question x_i, after which a linear transformation θ produces the question embedding $q_i = \theta\big(\phi(x_i)\big)$. For each question, we define the set of pruned-head models as $\mathcal{M} = \{1, \ldots, LH\}$ and partition this set into a positive subset $\mathcal{M}_i^+ = \{j : z_{ij} = 1\}$ containing heads that result in correct answers to question i, and a negative subset $\mathcal{M}_i^- = \{j : z_{ij} = 0\}$.

To further quantify the correspondence between different pruned-head models, we introduce a similarity measure between any two heads j and k as follows:

$$s_{jk} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(z_{ij} = z_{ik}) \in [0, 1],$$

which computes the fraction of questions for which both heads produce the same outcome. We jointly optimize the transformation θ and the head embeddings $\mathcal{V}$ with the following loss function:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} -\log\left(\frac{\sum_{j\in\mathcal{M}_i^+}\exp\Big(-\|q_i - v_j\|_2^2\Big)}{\sum_{j=1}^{LH}\exp\Big(-\|q_i - v_j\|_2^2\Big)}\right) - \lambda\sum_{j<k} s_{jk}\,\|v_j - v_k\|_2^2. \tag{1}$$

Our joint optimization loss $\mathcal{L}$ consists of two components. The first term aims to align the question embedding q_i with the embeddings v_j of those pruned-head models that yield the correct answer. The loss employs a softmax formulation for each question, over the negative squared Euclidean distances between q_i and the head embeddings. This loss encourages the model to produce a lower distance, and thus a higher similarity, for those heads in the positive set $\mathcal{M}_i^+$, which results in a correct answer.

The second term of the loss addresses the need for diversity among head embeddings. By summing the pairwise distances weighted by the similarity measure s_{jk}, it acts as a regularizer that encourages distinctiveness between head embeddings, especially for those heads that tend to yield similar outcomes. This term effectively pushes similar heads apart in the embedding space to reduce redundancy and promote a richer, more discriminative representation. Overall, combining these two terms helps align the question representation with high-performing pruned models while maintaining a diverse set of head embeddings.

Inference. At inference time, we obtain the embedding for the new question x by feedforwarding $q = \theta(\phi(x))$. The optimal head to prune for a given test question is identified by selecting the head whose embedding minimizes the squared Euclidean distance to the question embedding, i.e.,

$$j^\star = \arg\min_{j\in\mathcal{M}} \|q - v_j\|_2^2.$$

We integrate best-of-N reasoning into our framework to enhance the performance by selecting the top-N pruned-head configurations based on their proximity to q. We obtain candidate answers via greedy decoding with a temperature of 0.

Overall, our unified methodology demonstrates that pruning the attention heads, guided by a contrastive alignment strategy, can improve model performance. Thus, the SPRINT framework identifies which heads benefit from pruning and supports a

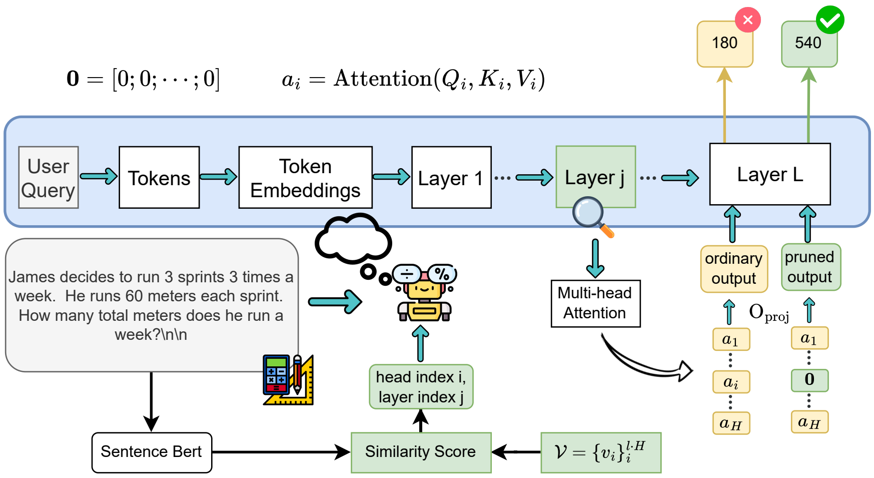

Figure 3: Schematic overview of our head pruning framework.

dynamic simulation approach to generate optimal predictions.

5 Numerical Experiments

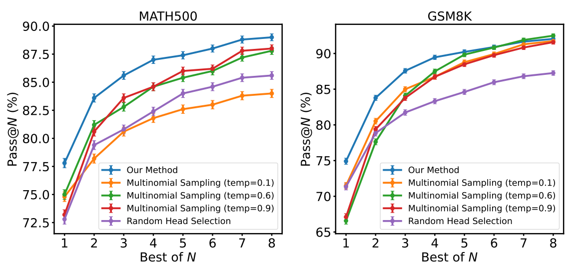

Figure 4: Average Pass@N on MATH500 and GSM8K dataset.

Datasets. For our experiments, we use the MATH and GSM8K datasets. We provide more details in the appendix.

Base Models. As a proof of concept, we select `Qwen2.5-Math-1.5B-Instruct` as our generation model. This choice is motivated by its strong empirical gains observed in Figure 2. All experiments are performed using the original, unquantized version of this model. We consider five layers {5, 10, 15, 20, 25} for pruning.

Performance metrics. To assess performance, we use the Pass@N metric. Specifically, each method is used to sample N candidate answers, and the accuracy is computed based on the best answer as determined by an oracle.

Baselines. We compare our methods against two baselines: multinomial sampling, commonly used for Best-of-N Reasoning, and a random head selection strategy. We perform Best-of-N evaluations using three temperature values for multinomial sampling: 0.1, 0.6, and 0.9. For the random head selection strategy, we use a greedy approach to identify the heads that solve the most samples in the training set. During testing, we randomly select N heads and measure Pass@N.

The project's repository is available at https://github.com/HieuNT91/attention_pruning.

5.1 Quantitative Results

Figure 4 summarizes our primary findings. Our method consistently outperforms the baselines for nearly all values of N across the MATH and GSM8K datasets. Notably, the greatest improvements are observed at lower values $N \in \{1, 2, 3, 4\}$, while the performance gains become more gradual as N increases.

6 Conclusion

In this work, we demonstrated that pruning attention heads, a technique traditionally viewed as harmful, can enhance the reasoning capabilities of transformer-based language models. Our study revealed that the selective removal of specific heads mitigates the propagation of redundant or noisy signals and enhances overall model performance. We developed a contrastive learning framework to harness this phenomenon that dynamically selects the optimal head and layer to prune at inference time. Experimental results confirm that our approach significantly outperforms traditional Best-of-N methods and random head selection strategies.

Limitations

Due to the space limitations of a short paper, our work does not experiment with a more diverse range of model architectures and sizes. Our work primarily focuses on the mathematical domain, but it can be extended to other scientific domains that require LMs' reasoning.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Edward Beeching, Lewis Tunstall, and Sasha Rush. 2024. Scaling test-time compute with open models.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning. *arXiv preprint arXiv:2412.09078*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Ngoc Bui, Hieu Trung Nguyen, Shantanu Kumar, Julian Theodore, Weikang Qiu, Viet Anh Nguyen, and Rex Ying. 2025. Mixture-of-personas language models for population simulation. *arXiv preprint arXiv:2504.05019*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. 2024. Learning how hard to think: Input-adaptive allocation of LM computation. *arXiv preprint arXiv:2410.04707*.

John Doe, Jane Smith, and Alan Turing. 2023. Natural language processing in virtual assistants: Current approaches and challenges. In *Proceedings of the IEEE Conference on Artificial Intelligence*, pages 122–128. IEEE.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*.

Maria Garcia, Li Wei, and Ahmed El-Sayed. 2023. Ethical challenges in the development of virtual assistants powered by large language models. *Electronics*, 12(14):3170.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rStar-Math: Small LLMs can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Advances in Neural Information Processing Systems*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Chonghe Jiang, Bao Nguyen, Anthony Man-Cho So, and Viet Anh Nguyen. 2025. Probe-free low-rank activation intervention. *arXiv preprint arXiv:2502.04043*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024a. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024b. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2023a. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, and 1 others. 2023b. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Bao Nguyen, Binh Nguyen, Duy Nguyen, and Viet Anh Nguyen. 2025a. Risk-aware distributional intervention policies for language models. *arXiv preprint arXiv:2501.15758*.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025b. Multi-attribute steering of language models via targeted intervention. *arXiv preprint arXiv:2502.12446*.

Hieu Trung Nguyen, Bao Nguyen, Binh Nguyen, and Viet Anh Nguyen. 2025c. Task-driven layerwise additive activation intervention. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 506–513, Albuquerque, New Mexico. Association for Computational Linguistics.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, and 1 others. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, and 1 others. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Emre Sezgin. 2024. Redefining virtual assistants in health care: The future with large language models. *Journal of Medical Internet Research*, 26:e53225.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2019. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*.

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. *Advances in Neural Information Processing Systems*, 35:32353–32368.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah D Goodman, Christopher D Manning, and Christopher Potts. 2024. pyvene: A library for understanding and improving pytorch models via interventions. *arXiv preprint arXiv:2403.07809*.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte Carlo tree search boosts reasoning via iterative preference learning. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-MCTS*: LLM self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*.

A Implementation details.

Datasets. The MATH dataset contains 12,500 math problems, with 12,000 problems designated for training and 500 for testing. Similarly, the GSM8K dataset comprises 8,792 math problems, with 7,473 used for training and 1,319 for testing. In our experiments, we subsample 1,500 training examples from the respective training sets. The head embeddings $\mathcal{V}$ are then trained using the subsampled training set. Finally, we evaluate performance on the test splits of the aforementioned datasets.

B Empirical Observation on the Impact of Attention Head Pruning.

Figure 5 shows that pruning an attention head in `Qwen-2.5-1.5B Math Instruct` model mostly leads to improvement in accuracy@1 on the MATH500 dataset.

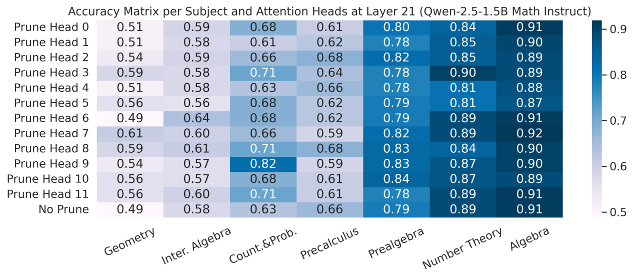

Figure 5: Impact of individual head pruning on model performance.

Table 1: We report Pass@1 scores for single head pruning across different layers of `Meta-Llama-3-8B-Instruct`, organized by question statement subjects.

	Algebra				Counting & Probability				Geometry				Intermediate Algebra				Number Theory				Prealgebra				Precalculus			
Layer	5	10	15	31	5	10	15	31	5	10	15	31	5	10	15	31	5	10	15	31	5	10	15	31	5	10	15	31
Prune Head 0	0.35	0.37	0.32	0.32	0.11	0.11	0.13	0.16	0.2	0.24	0.2	0.22	0.13	0.14	0.14	0.16	0.19	0.19	0.21	0.18	0.4	0.38	0.41	0.39	0.14	0.12	0.14	0.12
Prune Head 1	0.31	0.37	0.32	0.34	0.11	0.11	0.08	0.08	0.2	0.22	0.22	0.24	0.14	0.13	0.14	0.13	0.19	0.24	0.19	0.24	0.41	0.37	0.37	0.4	0.11	0.14	0.11	0.11
Prune Head 2	0.33	0.32	0.32	0.34	0.16	0.13	0.11	0.11	0.22	0.15	0.22	0.2	0.14	0.13	0.15	0.15	**0.27**	0.18	0.24	0.18	0.34	0.34	0.41	0.41	0.11	0.09	0.09	0.12
Prune Head 3	0.35	0.31	0.34	0.33	0.08	0.11	0.11	0.13	0.17	0.22	0.24	0.22	0.13	0.1	**0.16**	0.16	0.19	0.16	0.21	0.19	0.38	**0.48**	0.35	0.34	0.09	**0.18**	0.11	0.09
Prune Head 4	0.31	0.34	0.3	0.33	0.11	0.11	0.16	0.11	0.22	0.22	0.22	0.22	0.14	0.13	0.15	0.14	0.15	0.21	0.23	0.19	**0.45**	0.39	0.44	0.38	0.12	0.07	0.11	0.11
Prune Head 5	0.33	0.33	0.36	0.33	0.16	0.08	0.13	0.11	0.2	0.24	0.24	0.24	0.12	0.12	0.1	0.11	0.16	**0.27**	0.23	0.21	0.39	0.37	0.4	**0.43**	0.11	0.12	0.11	0.09
Prune Head 6	0.35	0.35	0.38	0.32	0.13	0.11	0.13	0.13	0.2	0.2	0.22	0.24	0.1	0.13	0.1	0.14	0.26	0.18	0.23	0.19	0.38	0.43	0.39	0.39	0.09	**0.18**	0.12	0.11
Prune Head 7	0.35	0.34	0.31	0.31	0.11	0.05	0.13	0.13	0.17	0.24	0.17	0.22	0.11	0.11	0.12	0.16	0.21	0.21	0.21	0.19	0.39	0.4	0.43	0.35	0.07	0.09	0.14	0.11
Prune Head 8	0.35	0.37	0.31	0.34	0.13	0.18	0.11	0.16	0.22	0.24	0.22	0.2	0.11	0.12	0.12	0.13	0.23	0.21	0.13	0.21	0.41	0.38	0.34	0.4	0.12	0.12	**0.18**	0.11
Prune Head 9	0.35	0.31	0.36	0.32	0.13	0.08	0.08	0.13	0.22	0.17	0.24	0.22	0.13	0.1	0.13	0.16	0.21	0.23	**0.26**	0.21	0.41	0.39	0.34	0.39	0.07	0.11	0.16	0.11
Prune Head 10	0.33	0.37	0.38	0.34	0.16	0.08	0.08	0.13	0.24	0.2	0.17	0.24	0.14	0.09	0.14	0.15	0.19	0.21	0.21	0.21	0.4	0.4	0.39	0.4	0.11	**0.18**	0.12	0.12
Prune Head 11	0.33	0.34	0.32	0.33	0.11	0.05	**0.18**	0.13	0.22	0.24	0.27	0.2	**0.16**	0.12	0.13	0.16	0.21	0.18	0.18	0.19	0.37	0.38	0.37	0.37	0.14	0.12	0.12	0.11
Prune Head 12	0.35	0.34	0.33	0.35	0.11	0.13	**0.18**	0.13	0.27	0.22	0.2	0.24	0.14	0.12	0.14	0.12	0.24	0.19	0.19	0.21	0.38	0.44	0.43	0.39	0.12	0.12	0.12	0.07
Prune Head 13	0.35	0.35	0.27	0.35	0.13	0.05	**0.18**	**0.18**	0.15	0.24	0.1	0.2	**0.16**	0.12	0.11	0.14	0.23	0.21	0.16	0.21	0.41	0.41	0.28	**0.43**	0.11	0.12	0.12	0.11
Prune Head 14	0.35	0.38	0.32	0.27	0.11	**0.24**	0.11	0.08	0.2	0.2	0.17	0.17	0.14	0.13	0.13	0.12	0.23	0.23	0.16	0.18	0.38	0.38	0.3	0.37	0.12	0.12	0.11	**0.14**
Prune Head 15	0.3	0.31	0.35	0.32	0.13	0.11	0.16	0.16	0.15	0.17	0.2	0.22	0.1	**0.16**	0.09	0.13	0.23	0.21	0.23	0.19	0.41	0.35	0.35	0.41	0.14	0.11	0.11	0.07
Prune Head 16	0.35	0.35	0.33	0.33	0.13	0.11	0.13	0.13	0.22	0.24	0.22	0.24	0.13	0.14	0.15	0.15	0.18	0.18	0.21	0.21	0.39	0.4	0.44	0.39	**0.18**	0.11	0.12	0.11
Prune Head 17	0.33	0.36	0.35	0.35	0.13	0.13	0.13	0.11	0.17	0.2	0.24	0.22	0.12	0.11	0.14	0.15	0.13	0.18	0.21	0.16	0.34	0.37	0.39	0.4	0.12	0.12	0.12	0.11
Prune Head 18	0.35	0.34	0.36	0.36	0.11	0.13	0.05	0.08	0.2	0.2	**0.29**	0.17	0.14	0.12	0.11	0.15	0.24	0.24	0.23	0.23	0.35	0.37	0.41	0.37	0.16	0.14	0.11	0.11
Prune Head 19	0.32	0.36	0.34	0.33	0.11	0.13	0.13	0.13	0.22	0.22	0.2	**0.27**	0.1	0.12	0.11	0.15	0.16	0.21	0.19	0.19	0.35	0.37	0.39	0.35	0.07	0.16	0.12	0.11
Prune Head 20	0.31	0.37	0.34	0.35	0.13	0.11	0.05	0.16	0.22	0.24	0.22	0.2	0.12	0.14	0.12	0.15	0.21	0.21	0.16	0.21	0.38	0.37	0.37	0.38	0.11	0.09	0.11	0.12
Prune Head 21	0.32	0.32	0.33	0.34	0.13	0.13	0.16	0.11	0.24	0.15	0.24	0.24	0.14	0.09	**0.16**	0.15	0.23	0.11	0.16	0.21	0.4	0.38	0.39	0.38	0.11	0.14	0.11	0.11
Prune Head 22	0.35	0.37	0.35	0.35	0.05	0.13	0.16	0.08	0.22	0.2	0.2	0.24	0.14	0.14	**0.16**	0.15	0.15	0.19	0.21	0.21	0.38	0.4	0.39	0.39	0.12	0.14	0.14	0.11
Prune Head 23	0.35	**0.40**	0.33	0.35	0.13	0.13	0.11	0.16	0.2	0.22	0.22	0.22	0.15	0.13	0.13	**0.18**	0.23	0.18	0.23	0.23	0.39	0.38	0.41	0.4	0.11	0.14	0.12	0.11
Prune Head 24	0.36	0.33	**0.39**	0.32	**0.21**	0.13	0.05	0.08	**0.29**	0.27	0.2	0.1	0.12	0.14	0.14	0.13	0.18	0.24	0.18	**0.29**	0.38	0.35	0.4	0.41	0.11	0.11	0.09	0.12
Prune Head 25	0.34	0.32	0.33	0.34	0.11	0.13	0.08	0.11	0.22	0.2	0.2	0.2	0.09	0.13	0.12	0.14	0.21	0.21	0.18	0.21	0.38	0.44	**0.46**	**0.43**	0.11	0.11	0.12	0.11
Prune Head 26	0.32	0.27	0.35	**0.40**	0.11	0.16	0.13	0.08	0.2	0.24	0.2	0.17	0.14	0.12	0.12	0.14	0.18	0.18	0.16	0.23	0.4	0.41	0.39	0.4	0.12	0.14	0.14	0.12
Prune Head 27	0.34	0.31	0.33	0.34	0.08	0.11	0.08	0.13	0.22	**0.29**	0.24	**0.27**	0.14	0.14	**0.16**	0.14	0.19	0.26	0.18	0.26	0.38	0.35	0.41	0.39	0.11	0.09	0.09	0.07
Prune Head 28	0.32	0.31	0.35	0.35	0.11	0.21	0.13	0.13	0.2	0.22	0.24	0.2	0.11	0.1	0.15	0.14	0.21	0.19	0.16	0.21	0.38	0.4	0.39	0.38	0.12	0.12	0.11	0.09
Prune Head 29	**0.37**	0.38	0.34	0.33	0.13	0.13	0.08	0.11	**0.29**	0.2	0.22	0.24	0.14	0.15	0.15	0.15	0.21	0.16	0.19	0.21	0.41	0.43	0.39	0.4	0.07	0.14	0.14	0.11
Prune Head 30	0.35	0.34	0.35	0.33	0.13	0.16	0.16	0.05	0.2	0.17	0.24	0.24	0.13	0.13	0.14	0.15	0.19	0.23	0.19	0.21	0.39	0.38	0.37	0.38	0.11	0.14	0.16	0.11
Prune Head 31	0.31	0.33	0.37	0.33	0.13	0.13	0.16	0.16	0.27	0.24	0.2	0.24	0.14	0.12	0.13	0.14	0.19	0.16	0.23	0.19	0.4	0.4	0.39	0.39	0.11	0.14	0.11	0.11
No Prune	0.33	0.33	0.33	0.33	0.11	0.11	0.11	0.11	0.24	0.24	0.24	0.24	0.15	0.15	0.15	0.15	0.21	0.21	0.21	0.21	0.39	0.39	0.39	0.39	0.11	0.11	0.11	0.11

Table 2: We report Pass@1 scores for single head pruning across different layers of `Qwen2.5-Math-1.5B-Instruct`, organized by question statement subjects.

	Algebra				Counting & Probability				Geometry				Intermediate Algebra				Number Theory				Prealgebra				Precalculus			
Layer	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27
Prune Head 0	0.92	0.9	0.91	0.91	0.63	0.63	**0.79**	0.66	0.56	0.49	0.54	0.51	0.59	**0.62**	0.55	**0.62**	0.82	0.87	**0.89**	0.87	**0.82**	0.82	0.79	0.78	0.62	0.59	0.57	0.68
Prune Head 1	0.92	0.92	0.92	0.92	0.71	0.68	0.63	0.63	0.51	0.56	0.59	0.51	0.6	0.52	0.53	0.58	0.84	0.85	0.85	**0.90**	**0.82**	0.77	0.78	0.79	0.62	0.62	0.64	0.66
Prune Head 2	**0.93**	0.91	0.91	0.91	**0.74**	0.66	0.66	**0.71**	**0.59**	0.54	**0.63**	0.54	**0.61**	0.58	0.54	0.59	**0.92**	**0.89**	0.84	0.89	**0.82**	**0.83**	0.77	0.79	0.61	0.62	0.61	0.68
Prune Head 3	0.06	0.91	0.92	0.9	0.08	0.71	0.61	**0.71**	0.07	0.56	0.61	0.49	0.07	0.6	**0.59**	0.6	0.1	0.85	**0.89**	**0.90**	0.12	0.79	**0.83**	0.79	0.09	0.62	0.66	0.68
Prune Head 4	0.89	0.92	**0.93**	0.9	0.71	0.66	0.66	0.68	**0.59**	0.54	0.54	0.51	0.59	0.6	0.55	0.59	0.85	0.82	0.87	0.89	**0.82**	0.78	0.77	0.78	0.62	0.62	0.62	0.68
Prune Head 5	0.56	0.91	0.9	0.91	0.5	0.61	0.61	0.66	0.46	**0.59**	0.54	**0.56**	0.37	0.58	0.53	0.59	0.47	**0.89**	0.81	0.84	0.54	**0.83**	0.76	0.77	0.41	0.64	**0.68**	0.66
Prune Head 6	0.81	0.92	0.9	0.9	0.61	0.66	0.63	0.63	0.56	0.56	0.59	0.51	0.51	0.59	**0.59**	0.57	0.71	0.82	0.82	0.89	0.73	0.82	0.74	0.76	0.54	0.66	0.66	0.64
Prune Head 7	0.91	0.92	0.9	0.9	0.63	0.61	0.63	0.66	0.56	0.56	0.54	0.54	0.58	0.6	0.56	0.59	0.85	0.87	**0.89**	0.87	0.78	0.79	0.8	0.78	0.64	**0.68**	0.55	0.62
Prune Head 8	0.86	**0.93**	0.9	**0.93**	0.63	**0.74**	0.68	0.68	**0.59**	0.49	0.56	0.54	0.53	0.56	**0.59**	0.58	0.82	0.85	0.87	0.85	0.79	0.8	0.82	**0.80**	0.62	0.61	0.62	0.64
Prune Head 9	**0.93**	0.91	0.91	0.91	0.66	0.66	0.68	0.66	0.56	0.49	0.54	0.49	0.57	0.54	0.52	0.57	0.9	0.84	0.85	0.89	0.78	0.82	0.8	0.79	0.64	0.66	0.66	0.66
Prune Head 10	0.92	0.92	0.92	0.89	0.68	0.66	0.68	0.66	0.54	0.51	0.56	0.51	0.55	0.57	0.56	0.57	0.85	0.85	0.84	0.87	0.8	0.8	0.77	0.78	**0.66**	0.62	0.62	**0.70**
Prune Head 11	0.9	0.92	0.91	0.9	0.68	0.66	0.71	0.66	0.51	0.56	0.56	0.49	0.6	0.61	0.55	0.6	0.87	0.84	0.87	0.87	0.78	0.8	0.77	**0.80**	**0.66**	0.62	0.61	0.66
No Prune	0.91	0.91	0.91	0.91	0.63	0.63	0.63	0.63	0.49	0.49	0.49	0.49	0.58	0.58	0.58	0.57	0.89	**0.89**	**0.89**	0.87	0.79	0.79	0.79	0.79	**0.66**	0.66	0.66	0.68

Table 3: We report Pass@1 scores for single head pruning across different layers of `Qwen2.5-Math-7B-Instruct`, organized by question statement subjects.

	Algebra				Counting & Probability				Geometry				Intermediate Algebra				Number Theory				Prealgebra				Precalculus			
Layer	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27
Prune Head 0	0.96	0.96	0.97	**0.97**	**0.84**	0.82	0.84	**0.84**	0.68	0.68	0.59	0.66	**0.68**	0.67	0.64	0.62	0.9	0.95	0.95	**0.97**	0.9	0.88	0.9	0.88	0.71	0.77	0.7	**0.80**
Prune Head 1	**0.97**	0.96	0.97	**0.97**	**0.84**	0.79	0.84	0.82	0.68	0.66	0.66	0.66	0.63	0.65	0.63	0.62	0.95	0.95	0.97	0.95	0.89	0.89	0.89	0.88	0.73	0.75	0.75	0.77
Prune Head 2	**0.97**	0.96	**0.98**	**0.97**	**0.84**	**0.84**	**0.87**	**0.84**	0.66	0.66	0.66	0.66	0.65	0.64	0.64	0.65	0.92	0.95	0.92	0.95	0.9	0.89	0.9	0.87	0.75	0.73	0.71	0.77
Prune Head 3	0.36	0.97	0.97	**0.97**	0.5	0.82	0.82	0.82	0.29	0.61	0.61	0.66	0.25	0.65	0.68	0.62	0.29	**0.97**	**1.00**	0.95	0.34	0.89	0.88	0.88	0.2	0.77	0.73	0.75
Prune Head 4	**0.97**	0.97	0.95	**0.97**	**0.84**	0.82	0.82	0.82	0.68	0.66	**0.68**	0.66	0.61	0.66	0.63	0.62	0.95	0.94	0.92	0.95	0.89	**0.90**	0.89	**0.90**	0.75	0.73	0.75	0.75
Prune Head 5	0.96	0.95	0.96	**0.97**	0.82	0.82	0.84	0.82	**0.73**	0.63	0.63	0.68	0.63	0.67	0.65	0.62	0.95	0.94	0.97	0.92	0.89	**0.90**	0.88	0.88	0.73	0.75	0.71	0.75
Prune Head 6	**0.97**	0.96	0.96	**0.97**	**0.84**	0.82	0.82	0.82	0.63	0.68	0.63	0.66	0.66	0.61	0.67	0.63	0.92	0.94	0.95	0.95	0.88	0.89	0.9	0.89	0.68	0.73	0.77	0.79
Prune Head 7	0.96	0.97	0.97	**0.97**	0.79	0.82	0.82	0.82	0.61	0.66	0.63	0.73	0.64	0.64	0.66	0.62	0.95	0.95	0.95	0.95	0.89	0.89	0.89	0.89	0.73	**0.79**	0.75	0.71
Prune Head 8	**0.97**	0.97	0.97	**0.97**	0.82	**0.84**	0.84	**0.84**	0.68	0.66	**0.68**	0.68	0.65	0.63	0.65	0.63	**0.97**	0.95	0.94	**0.97**	0.89	0.88	0.88	0.88	0.75	0.75	0.7	0.71
Prune Head 9	0.96	0.97	0.96	**0.97**	**0.84**	**0.84**	0.82	**0.84**	0.61	0.66	0.59	**0.76**	0.63	0.65	0.64	**0.67**	0.94	0.95	0.97	0.95	0.89	0.88	0.9	0.89	0.77	0.75	0.77	0.73
Prune Head 10	0.95	0.97	0.96	0.96	**0.84**	0.82	0.84	**0.84**	0.66	0.63	0.63	0.71	0.64	0.63	0.65	0.64	0.94	0.95	0.94	0.95	0.89	0.88	0.89	0.88	0.73	0.75	0.71	0.75
Prune Head 11	**0.97**	0.97	0.97	**0.97**	0.82	0.82	0.82	**0.84**	0.63	**0.71**	0.63	0.73	0.64	0.65	**0.69**	0.63	**0.97**	0.94	0.95	0.95	0.9	0.88	0.9	0.88	**0.79**	0.71	0.77	0.75
Prune Head 12	**0.97**	0.97	**0.98**	**0.97**	0.82	0.82	**0.87**	**0.84**	0.66	0.66	**0.68**	0.71	0.66	0.63	0.67	0.64	0.95	0.95	0.92	**0.97**	0.88	0.89	0.87	0.88	0.77	0.77	0.75	0.73
Prune Head 13	**0.97**	0.97	**0.98**	**0.97**	0.82	0.82	0.82	**0.84**	0.61	0.68	**0.68**	0.68	0.6	0.64	0.63	0.64	0.95	0.95	0.94	**0.97**	0.88	0.88	**0.91**	0.88	0.73	0.75	**0.79**	0.75
Prune Head 14	0.96	0.97	0.96	**0.97**	**0.84**	0.82	0.84	0.82	0.63	0.61	0.66	0.66	0.67	0.66	0.63	0.61	0.94	0.95	0.95	0.95	0.89	0.88	0.89	0.85	0.77	0.73	0.71	0.75
Prune Head 15	**0.97**	0.97	0.94	**0.97**	0.79	0.82	0.84	0.82	**0.73**	0.68	0.63	0.61	0.66	0.66	0.64	0.63	**0.97**	0.94	0.94	0.95	0.9	0.88	**0.91**	0.88	0.77	0.71	0.75	0.75
Prune Head 16	0.96	0.96	0.97	**0.97**	0.82	**0.84**	0.82	**0.84**	0.68	0.63	**0.68**	0.66	0.67	0.62	0.64	0.63	0.9	0.95	0.97	0.95	0.88	0.89	0.89	0.87	0.73	0.75	0.75	0.75
Prune Head 17	0.96	0.96	**0.98**	**0.97**	0.82	0.82	0.82	0.82	0.66	0.63	**0.68**	0.61	0.65	0.66	0.64	0.63	0.92	0.94	0.92	0.95	0.87	0.88	0.89	0.88	0.75	0.75	0.75	0.75
Prune Head 18	0.96	0.97	0.97	**0.97**	0.82	0.82	**0.87**	0.82	0.61	0.63	0.66	0.66	0.67	0.6	0.61	0.63	0.94	0.94	0.95	0.95	0.89	0.89	0.88	0.88	0.77	0.77	0.77	0.75
Prune Head 19	0.96	0.96	0.97	**0.97**	0.82	0.82	0.79	0.82	0.63	0.68	0.66	0.63	0.67	0.64	0.65	0.64	0.92	0.95	0.95	0.95	0.87	0.89	0.89	**0.90**	0.73	0.71	0.71	0.75
Prune Head 20	0.96	0.97	0.96	**0.97**	**0.84**	0.82	0.82	0.82	0.63	0.61	0.66	0.68	0.65	0.62	0.65	0.62	0.92	0.95	0.94	0.95	0.88	**0.90**	0.89	0.88	0.75	0.71	0.73	0.75
Prune Head 21	**0.97**	0.96	0.97	**0.97**	0.79	**0.84**	0.82	**0.84**	0.63	0.59	0.66	0.66	0.64	0.62	0.66	0.61	0.9	0.95	0.95	0.95	**0.91**	0.88	0.89	0.88	0.73	0.75	0.71	0.71
Prune Head 22	**0.97**	0.97	0.96	**0.97**	**0.84**	**0.84**	0.82	**0.84**	0.71	0.66	**0.68**	0.66	0.62	0.63	0.62	0.61	0.95	0.95	0.97	0.94	0.9	0.89	0.88	0.88	0.73	0.73	0.75	0.75
Prune Head 23	0.96	0.97	**0.98**	**0.97**	**0.84**	**0.84**	0.82	0.82	0.63	0.66	0.61	0.63	0.65	0.64	0.65	0.64	0.95	0.94	0.94	0.95	0.9	0.89	0.9	0.88	0.73	0.73	0.77	0.75
Prune Head 24	**0.97**	0.96	0.96	**0.97**	0.82	0.82	0.79	**0.84**	**0.73**	0.63	0.66	0.63	0.65	0.62	0.66	0.62	0.94	0.95	0.9	0.95	0.89	0.88	0.9	0.89	0.73	0.77	0.77	0.77
Prune Head 25	**0.97**	0.97	0.96	**0.97**	**0.84**	0.82	0.79	0.82	0.63	**0.71**	**0.68**	0.66	0.64	**0.68**	0.61	0.61	0.95	0.94	0.94	0.95	0.88	0.89	**0.91**	**0.90**	0.75	0.73	0.73	0.77
Prune Head 26	**0.97**	**0.98**	**0.98**	**0.97**	0.79	0.82	0.79	**0.84**	0.71	0.68	**0.68**	0.68	0.62	0.67	0.62	0.61	0.95	0.95	0.95	0.94	0.88	0.88	0.88	0.89	0.73	**0.79**	0.75	0.77
Prune Head 27	**0.97**	0.97	0.97	**0.97**	**0.84**	0.79	0.82	0.82	0.66	0.66	0.66	0.61	0.62	0.65	0.64	0.63	0.95	0.95	0.95	0.94	0.88	**0.90**	**0.91**	0.88	0.73	0.73	0.68	0.75
No Prune	**0.97**	0.97	0.97	**0.97**	0.82	0.82	0.82	0.82	0.66	0.66	0.66	0.66	0.62	0.62	0.62	0.62	0.95	0.95	0.95	0.95	0.88	0.88	0.88	0.88	0.75	0.75	0.75	0.75

Table 4: We report Pass@1 scores for single head pruning across different layers of `Qwen2.5-7B-Instruct`, organized by question statement subjects.

	Algebra				Counting & Probability				Geometry				Intermediate Algebra				Number Theory				Prealgebra				Precalculus			
Layer	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27	0	5	15	27
Prune Head 0	0.55	0.94	0.95	0.95	0.5	0.74	**0.76**	0.71	0.44	0.56	0.56	0.59	0.38	0.55	0.54	0.57	0.34	0.85	0.85	0.87	0.49	0.83	0.84	0.87	0.36	0.59	0.61	0.59
Prune Head 1	0.85	0.94	0.93	0.94	0.61	0.71	0.74	0.71	0.49	0.63	0.56	0.61	0.51	0.57	0.53	0.59	0.82	0.87	0.87	0.87	0.73	0.84	0.83	0.87	**0.70**	0.62	0.59	0.61
Prune Head 2	0.94	0.94	0.95	0.94	0.68	0.71	0.71	0.71	0.63	0.61	**0.71**	**0.66**	0.58	0.56	0.54	0.58	0.82	0.84	0.81	0.85	0.87	0.84	0.83	0.87	0.61	0.61	0.61	0.64
Prune Head 3	0.02	0.94	0.92	0.94	0	**0.76**	**0.76**	0.74	0.05	0.61	0.68	0.61	0.04	0.58	**0.60**	0.6	0	0.85	0.84	0.85	0.01	0.84	0.84	0.87	0.02	0.61	0.62	0.61
Prune Head 4	0.94	0.94	0.94	0.94	0.71	0.68	0.74	0.71	0.61	0.59	0.56	0.61	0.59	0.58	0.54	0.59	**0.85**	0.82	0.79	0.85	0.87	0.84	0.83	0.85	0.62	0.57	0.64	0.61
Prune Head 5	0.95	0.95	0.93	0.94	0.71	0.74	0.74	0.71	0.59	0.61	0.61	0.56	0.58	0.54	0.57	0.6	**0.85**	0.84	0.84	0.82	0.87	0.82	0.84	0.85	0.62	0.62	0.62	0.61
Prune Head 6	0.63	0.95	0.91	0.95	0.53	**0.76**	0.66	0.71	0.54	0.59	0.56	0.61	0.44	0.57	0.53	0.6	0.45	**0.89**	0.84	0.87	0.65	0.85	0.83	0.85	0.39	0.62	0.64	0.62
Prune Head 7	**0.96**	0.93	0.91	0.94	**0.76**	0.74	**0.76**	0.74	**0.68**	0.61	0.54	0.59	0.55	0.57	0.59	0.58	0.84	0.87	0.82	0.87	0.84	0.85	0.85	0.84	0.64	0.64	0.61	0.64
Prune Head 8	0.94	0.95	0.94	0.94	0.74	0.74	0.74	0.74	0.63	0.63	0.59	0.61	0.58	0.58	0.58	0.58	0.82	0.87	0.84	0.84	0.85	0.85	0.85	0.85	0.55	0.57	0.59	0.59
Prune Head 9	0.95	0.95	0.94	**0.97**	0.68	0.74	0.68	0.71	0.56	0.66	0.56	0.59	0.58	0.55	0.57	0.57	**0.85**	0.85	**0.92**	0.87	0.87	0.87	**0.87**	0.85	0.61	0.57	0.62	0.64
Prune Head 10	0.91	0.94	0.94	0.94	0.61	0.74	0.68	0.76	0.54	0.61	0.63	0.61	**0.60**	0.6	0.55	0.56	0.82	0.85	0.77	0.85	0.82	0.85	0.8	0.8	0.61	0.61	0.64	0.55
Prune Head 11	0.87	0.93	0.95	0.94	0.71	0.74	**0.76**	0.74	0.59	0.61	0.56	0.61	0.57	0.56	0.55	0.57	0.77	0.87	0.89	0.82	0.83	0.84	0.84	0.84	0.52	0.62	0.68	0.61
Prune Head 12	0.94	0.94	0.94	**0.97**	0.74	0.71	0.74	0.74	0.63	0.59	0.59	0.61	0.56	0.59	0.57	0.57	0.84	0.85	0.89	0.87	0.87	0.84	0.82	0.82	0.68	0.61	0.57	0.57
Prune Head 13	0.95	0.95	0.93	0.94	0.71	0.71	0.71	**0.79**	0.61	0.63	0.61	0.59	0.57	0.6	**0.60**	0.54	**0.85**	**0.89**	0.87	0.85	0.87	0.88	0.83	0.83	0.62	0.59	0.55	0.55
Prune Head 14	0.94	0.93	0.94	0.94	0.71	0.71	0.68	0.76	0.63	0.56	0.61	0.59	0.54	0.55	0.55	0.58	0.77	0.84	0.85	0.84	0.82	0.85	0.84	0.87	**0.70**	0.61	0.64	0.61
Prune Head 15	0.81	0.94	0.94	0.95	0.68	0.74	**0.76**	0.71	0.54	0.56	0.61	0.61	0.52	0.56	0.58	0.59	0.77	0.84	0.85	0.82	0.82	0.88	0.85	**0.88**	0.59	0.57	0.64	0.59
Prune Head 16	**0.96**	0.94	0.94	0.94	0.68	0.68	**0.76**	0.74	0.66	0.61	0.63	0.59	0.58	0.61	0.54	0.59	0.84	0.82	0.87	0.84	0.88	0.87	0.82	**0.88**	0.59	0.62	0.62	0.62
Prune Head 17	0.93	0.95	0.94	0.94	0.71	0.71	0.74	0.76	0.59	0.59	0.63	0.59	0.54	0.55	**0.60**	**0.61**	0.81	0.84	0.87	0.85	**0.89**	0.87	0.82	0.82	0.62	0.59	0.61	0.61
Prune Head 18	0.94	0.94	0.94	0.94	0.74	0.71	0.74	0.71	0.66	0.61	0.56	0.61	0.56	0.56	0.58	0.56	**0.85**	0.84	0.85	0.87	0.87	0.85	0.83	0.84	0.64	0.59	0.66	0.61
Prune Head 19	0.94	**0.96**	0.92	0.94	0.71	**0.76**	**0.76**	0.71	0.63	0.56	0.66	0.59	0.56	0.55	0.53	0.58	0.82	0.87	0.84	0.85	0.87	0.79	**0.87**	0.85	0.66	0.57	**0.70**	0.61
Prune Head 20	0.95	0.94	0.94	0.94	0.74	0.74	0.74	0.71	0.66	0.59	0.59	0.61	**0.60**	0.57	0.57	0.58	0.81	0.87	0.82	0.84	0.84	0.82	0.84	**0.88**	0.61	0.62	0.66	0.61
Prune Head 21	0.95	**0.96**	**0.96**	0.96	**0.76**	0.74	0.74	0.71	0.61	0.61	0.61	0.61	0.56	0.58	0.58	0.6	**0.85**	0.85	0.85	0.87	0.85	0.85	0.84	0.87	0.61	0.62	0.62	0.61
Prune Head 22	0.44	**0.96**	0.93	0.94	0.24	**0.76**	0.74	0.71	0.22	0.59	0.66	0.59	0.12	0.58	0.55	**0.61**	0.26	0.85	0.87	0.84	0.52	**0.89**	0.83	0.84	0.07	0.62	0.64	0.61
Prune Head 23	0.11	0.94	0.92	0.94	0.21	0.74	0.63	0.71	0.17	0.66	0.68	0.59	0.04	**0.62**	0.49	0.58	0.06	0.84	0.79	0.87	0.27	0.85	0.82	0.87	0.04	0.61	0.59	0.59
Prune Head 24	0.83	0.94	0.93	0.96	0.61	0.74	0.71	0.71	0.56	0.66	0.63	**0.66**	0.51	0.6	0.56	0.6	0.77	0.85	0.84	0.85	0.68	0.87	0.84	0.84	0.57	0.62	0.55	0.62
Prune Head 25	0.02	0.95	0.95	0.95	0	**0.76**	0.74	0.66	0	0.63	0.66	0.63	0	0.55	**0.60**	0.59	0	**0.89**	0.81	0.85	0.02	0.84	0.83	0.85	0.02	**0.70**	0.59	**0.66**
Prune Head 26	0.5	0.95	0.93	0.95	0.47	0.71	0.71	0.71	0.37	0.54	0.54	0.56	0.25	0.58	**0.60**	0.56	0.56	0.85	0.89	**0.89**	0.63	0.88	0.85	0.85	0.21	0.55	0.64	**0.66**
Prune Head 27	0.83	**0.96**	0.94	0.95	0.68	0.71	0.71	0.63	0.46	**0.68**	0.59	0.56	0.49	0.6	0.52	0.59	0.69	0.85	0.82	0.82	0.63	0.84	0.84	0.85	0.61	0.59	0.59	0.64
No Prune	0.94	0.94	0.94	0.94	0.74	0.74	0.74	0.74	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.84	0.84	0.84	0.84	0.87	0.87	**0.87**	0.87	0.59	0.59	0.59	0.59