Layer Importance for Mathematical Reasoning is Forged in Pre-Training and Invariant after Post-Training

Aadim Nepal[1*] **Safal Shrestha**[1] **Anubhav Shrestha**[1] **Minwu Kim**[1] **Jalal Naghiyev**[2]
Ravid Shwartz-Ziv[3] **Keith Ross**[1]

[1] New York University Abu Dhabi
[2] Technical University of Munich
[3] NYU Center for Data Science

Abstract

Large language models improve at math after instruction tuning, reinforcement learning, or knowledge distillation. We ask whether these gains come from major changes in the transformer layers or from smaller adjustments that keep the original structure. Using layer-wise ablation on base and trained variants, we find that math reasoning depends on a few critical layers, which stay important across all post-training methods. Removing these layers reduces math accuracy by as much as 80%, whereas factual recall tasks only show relatively smaller drops. This suggests that specialized layers for mathematical tasks form during pre-training and remain stable afterward. As measured by Normalized Mutual Information (NMI), we find that near these critical layers, tokens drift from their original syntactic clusters toward representations aligned with tokens less syntactically related but potentially more useful for downstream task.

1 Introduction

The capabilities of large language models (LLMs) have been improved through different post-training methods such as instruction tuning and reinforcement learning with human feedback, knowledge distillation from a teacher model, and reinforcement learning with verifiable rewards (RLVR) [1, 2, 3, 4, 5]. Yet, the following question remains unexplored:

Do post-training methods fundamentally change how models process mathematical problems or just make small adjustments to existing structures?

In this paper, we investigate this question through layer ablation experiments on two model families, Llama-3.1-8B [6] and Qwen-2.5-7B [7], examining four variants for each: the pre-trained base model, an instruction-tuned model, a knowledge-distilled model, and an RLVR-trained model. We evaluate these models on two mathematical reasoning benchmarks, GSM8K [8] and MATH500 [9]. We find that post-training largely preserves the layer importance structure in mathematical reasoning: the set of critical layers - whose removal leads to substantial performance drop - remains largely invariant across all four model variants. Additionally, we conducted the same experiment on the TriviaQA [10] factual recall task. Here, we observed a different layer importance pattern: no specific critical layers were found. Instead, removing any layer results in a smaller, more consistent drop in performance.

[1] Code available at: `https://github.com/anonymous-xyz-anonymous/Ablation`.

[2] *Correspondence to: `aadim.nepal@nyu.edu`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

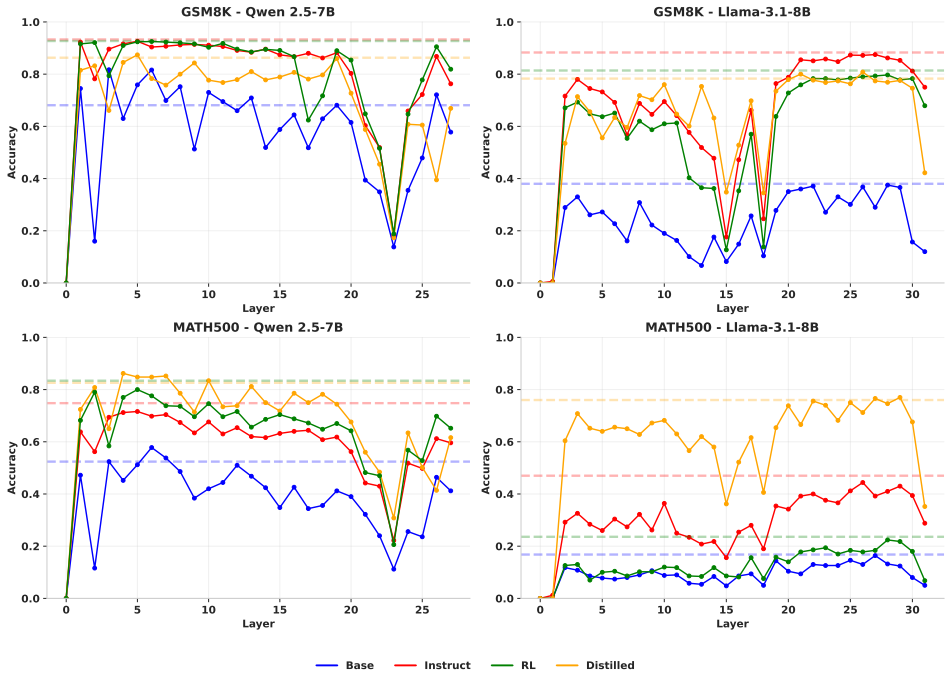

Figure 1: The plots show model accuracy (Y-axis) on GSM8K and MATH500 when a single transformer layer (X-axis) is zeroed out. The performance of all model variants drops substantially when specific layers are removed (layer 23 for Qwen, layers 15 and 18 for Llama), a pattern that remains consistent across different datasets and post-training methods. Dashed lines indicate the original, un-ablated performance.

Previous work [11, 12, 13, 14, 15] has mainly analyzed individual components of transformers—such as neurons, attention heads—but has not examined how post-training affects entire layer structures, or compared these effects between reasoning and non-reasoning [16, 17, 18, 19].

To investigate what happens at these critical layers, we apply a Normalized Mutual Information (NMI) analysis to track how token representations evolve across layers. Tokens are first clustered into families based on their initial semantics from the first layer. As the model approaches the critical layers, an increasing number of tokens drift across clusters, indicating that originally unrelated tokens begin to form new relationships at these points.

2 Related Work

Critical Layers and Ablation. Transformers respond unevenly to structural changes: some layers can be removed or reordered with little effect, while others are essential, especially for reasoning tasks [17]. Layer ablation has revealed "cornerstone layers," whose removal collapses performance [18], and "super weights," a small parameter set critical for fluency [19]. Zero-ablation studies further show that uncertainty and factuality share circuits [20]. Earlier work on BERT found layer specialization resembling an NLP pipeline [21, 22]. Other studies identified four inference stages across depth using ablation and probing [16], and many focus on smaller components—heads, neurons, or MLP blocks [11, 12, 13, 15, 14]. While most rely on zero-ablation, we use systematic layer-wise ablation to show that pre-training establishes critical layers for reasoning, and we extend this to study post-training effects. This complements findings that post-training preserves knowledge and truthfulness [23], but does not address layer-level stability across reasoning vs. non-reasoning tasks.

Representational Analysis with NMI. Prior work has analyzed internal representations to explain why certain layers matter, often using probing tasks or attention analysis [16, 24, 25]. While these approaches can reveal specific mechanisms, they require choosing properties to probe in advance and are less suited to generative tasks. Normalized Mutual Information (NMI), by contrast, compares

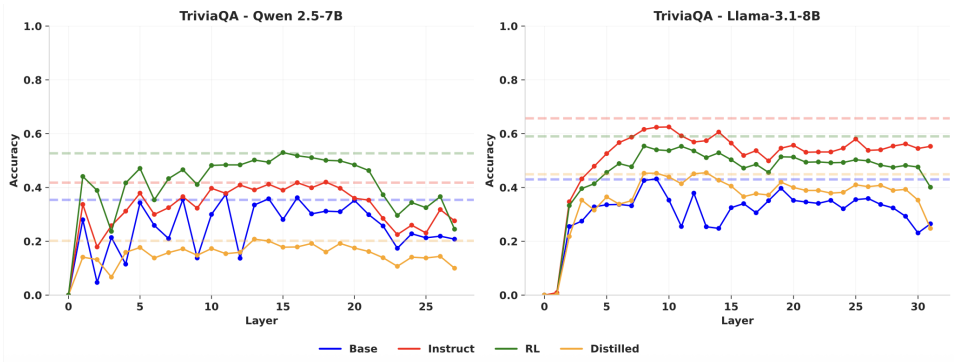

Figure 2: Layer ablation results on the TriviaQA factual recall task. The left plot shows performance for Qwen 2.5-7B models, and the right plot shows performance for Llama 3.1-8B models when individual layers are zeroed out. The X-axis represents the layer index (0-32), and the Y-axis shows the accuracy.

clustering structures across layers [26] and has shown, for example, that BERT encodes a linguistic hierarchy [27] and that representations shift during QA [22]. Intermediate layers in particular appear to hold rich, geometrically structured features [28, 29]. Our work uses NMI differently: instead of probing for specific linguistic features, we measure the extent of divergence from the input layer's clustering to highlight where semantic relationships form.

3 Experiments and Results

We perform systematic layer-wise zero ablation experiments to assess the importance of layers in the model. We choose the Qwen 2.5-7B and Llama-3.1-8B models as the primary base models for analysis. We take their respective post-trained variants as well for analysis. For math reasoning, we choose GSM8K [8] and MATH [9] and for factual recall, we choose TriviaQA [10]. Unlike past studies which primarily focus on classification benchmarks [30, 16], we opt for generative tasks for both math and non-math. This removes chances of random guesses and provides more accurate estimate of model abilities.

Furthermore, we track representational shifts across layers using Normalized Mutual Information (NMI), which measures the similarity between token clusterings at different layers. NMI ranges from 0 (independent) to 1 (identical), providing a simple way to quantify how far representations drift from their original input-layer clusters (Full details on the methodology including model details, ablation, and NMI are provided in Appendix A.3).

3.1 Ablation studies reveal stable critical layers

Our ablation studies reveal that mathematical reasoning in LLMs depends on a small set of stable critical layers. In Figure 1, for Qwen, performance drops sharply when ablating layer 23, while Llama is most sensitive at layers 15 and 18. Removing these layers still provides coherent responses but we find the model makes simple airthmetic mistakes when these layers are removed (Figure 6). Notably, these critical layers remain consistent after post-training: instruction tuning, distillation, and RLVR do not shift which layers are essential. We also find vulnerabilities at boundary layers: early layers are crucial for coherent language, consistent with "super-weight" findings [19]. For Qwen, ablating the first layer breaks fluency, while for Llama, the first two layers are critical. Some final layers in Llama also matter, reflecting architectural differences between models. Importantly, mathematical reasoning is far more sensitive to layer ablation than factual recall: removing critical layers causes accuracy collapses of 60–80%, whereas factual recall shows only modest, distributed declines of 10–30%. Interestingly, (as shared in Appendix A.8), Qwen's layer importance structures are similar across different model sizes as well, which could be because of their potentially shared pretraining schemes.

We observed a large drop at layer 2 of the Qwen base model. On closer inspection, most responses after removing layer 2 were incoherent. In contrast, removing layer 23 still produced coherent

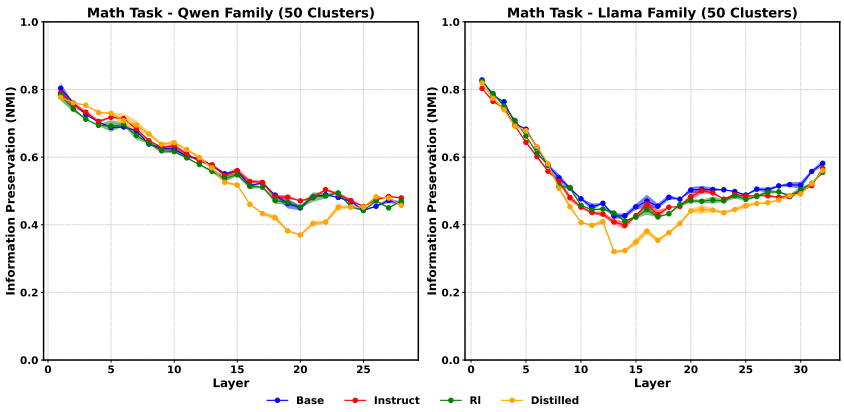

Figure 3: The plots show the NMI score (Y-axis) at each transformer layer (X-axis), calculated relative to the token clusters at Layer 0. The observed trends are robust to the number of clusters (k) used for the analysis, with similar results for k-values between 10 and 70. The choice of 50 here is arbitrary. Shaded region denotes standard deviation over 5 runs. For each run and each model, the 20 problems were selected randomly, so we are looking at over 100 math problems over 8 model families, which is about 800 problems in total.

responses (subsection A.6), but with many arithmetic errors, as shown in Figure 6. Finally, the plots in Figure 1 were averaged over hundreds of runs, and we confirm that the standard deviation at each point is minimal, so the results are stable and not due to random noise

3.2 Clustering Analysis

To gain some preliminary insight into the role of the critical layers, we performed a token clustering analysis using the procedure described in Appendix A.3.2. For each task, we constructed a single large-context prompt. For mathematical reasoning, the prompt concatenated 20 random problems and solutions from MATH500. Details are provided in Appendix A.7.

Figure 3 shows that for math tasks, the NMI score starts high and then drops in the middle-to-late layers, forming an "elbow" region. For Qwen models this appears between layers 20–25, and for Llama between layers 13–18. We use Layer 0 as the baseline because prior work shows that early layers tend to group tokens by syntactic type [22, 27]. A lower NMI score means that token clusters differ more from this baseline, indicating more mixing between token types. The layers identified as critical in our ablation experiments fall within this elbow region, suggesting a possible connection between token clustering changes and layer importance. In contrast, TriviaQA shows little change in NMI across layers (Figure 7). Token clusters remain close to the Layer 0 baseline, likely reflecting the lower token diversity in these prompts compared to math problems. This matches our ablation results, where no single critical layer was found for factual recall.

3.3 Qualitative Cluster Analysis of Token Representations

To complement our NMI analysis, we performed k-means clustering ($k = 10$) on hidden states at each layer for a representative mathematical reasoning prompt. This qualitative approach offers concrete illustrations of how token representations evolve through the network and helps interpret the quantitative trends captured by the NMI score.

3.3.1 Layer 0: Surface-Form Grouping

At the embedding layer (Layer 0), clusters are dominated by raw orthographic units and formatting tokens:

- **Symbols:** repeated tokens such as "`\frac`", "=", "−", and braces "{ { {" form distinct clusters.
- **Function words:** tokens such as "the", "of", "and", "by" co-cluster.

- **Special tokens:** <im_start|>|, `system`, `user`, and the initial prompt words appear together in a large mixed cluster.

This behavior reflects what NMI quantifies as high similarity to the Layer 0 baseline: tokens are grouped by surface type, with little semantic integration.

3.3.2 Layer 23: Emergence of Semantic Roles

By Layer 23 (the empirically identified critical layer for Qwen), clusters reflect semantically meaningful groupings aligned with the problem-solving process:

- **Problem setup (Cluster 3):** tokens representing the given equations, e.g. "$2x = 3y = -z$" and "$6x = -y = -4z$".
- **Parametric reformulation (Cluster 4):** "parametric", "form", "direction vector", and "line" co-occur.
- **Mathematical operations (Clusters 2 and 5):** fractions such as "$\frac{k}{2}$", "$\frac{1}{3}$", and symbols like "=" appear in equation-level context.
- **Conclusion (Cluster 7):** "angle", "dot product", "orthogonal", and "90" cluster together, corresponding to the final reasoning step.

3.3.3 Relation to NMI

The NMI analysis measures *how much* cluster structures diverge from Layer 0. The qualitative results here illustrate *what* those divergences look like. At critical layers such as Layer 23 in Qwen, where NMI reaches a local minimum, we see that tokens reorganize into semantically interpretable clusters that align with reasoning steps. This convergence of quantitative (NMI) and qualitative (clustering) evidence supports the interpretation that critical layers are precisely those where surface-level groupings are broken apart and recomposed into task-specific structures. In summary, from this qualitative token analysis, we observe that at early layers, tokens group mainly by syntactic meaning, for example brackets clustering with other brackets. Around the critical layers, these clusters reorganize into mixed groups that combine brackets with numbers or symbols. This supports the interpretation that tokens are drifting away from syntactic meaning and forming meaningful semantic relationships.

4 Limitations

We test on Qwen and Llama models at the 7B–8B scale, though broader model sizes and families should be explored for generalizability. We use NMI because it provides a high-level view of how token interactions evolve across layers, however, it cannot definitively establish reasoning. Also, other techniques like probing is better suited for classification tasks and requires deciding in advance what to probe for, while attention analysis typically focuses on individual prompts or tokens and thus fails to capture overall representational shifts [16, 24, 25]. For completeness, we also evaluated layer importance through residual contributions (Appendix A.11), but found no strong correlation with critical layers. This potentially suggests that hidden-state–based methods may not reliably identify task-relevant layers in generative mathematical reasoning, though further evidence is needed.

5 Future Work and Conclusion

We show that mathematical reasoning depends on a small set of critical layers that remain stable after post-training, whereas this behavior does not hold for non-reasoning tasks such as factual recall. Using additional token-clustering analysis, we provide an initial high-level explanation for why these layers may be particularly important for reasoning. Future work could explore targeted fine-tuning strategies; for instance, freezing all other layers and updating only these critical ones to improve mathematical reasoning efficiency. In addition, a more fine-grained mechanistic-interpretability analysis on these layers may shed light on why they are critical at a more granular level.

References

[1] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[3] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[5] OpenAI. Introducing OpenAI o1, 2025. Accessed: 2025-06-22.

[6] Meta. Introducing Llama 3.1: Our most capable models to date, 2024. Accessed: 2025-06-23.

[7] 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, 2024.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[10] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[11] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372, 2022.

[12] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[13] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, 2019.

[14] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, et al. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021.

[15] Nicholas Pochinkov, Ben Pasero, and Skylar Shibayama. Investigating neuron ablation in attention heads: The case for peak activation centering, 2024.

[16] Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference? *arXiv preprint arXiv:2406.19384*, 2024.

[17] Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. Transformer layers as painters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25219–25227, 2025.

[18] Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. Investigating layer importance in large language models. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen, editors, *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 469–479, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[19] Mengxia Yu, De Wang, Colorado Reed, and Alvin Wan. The super weight in large language models, 2025.

[20] Carter Teplica, Yixin Liu, Arman Cohan, and Tim G. J. Rudner. SCIURus: Shared circuits for interpretable uncertainty representations in language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12451–12469, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[21] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Association for Computational Linguistics*, 2019.

[22] Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1823–1832, 2019.

[23] Hongzhe Du, Weikai Li, Min Cai, Karim Saraipour, Zimin Zhang, Himabindu Lakkaraju, Yizhou Sun, and Shichang Zhang. How post-training reshapes llms: A mechanistic view on knowledge, truthfulness, refusal, and confidence, 2025.

[24] Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

[25] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

[26] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[27] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3662, Florence, Italy, July 2019. Association for Computational Linguistics.

[28] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2023.

[30] Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. *arXiv preprint arXiv:2403.19135*, 2024.

[31] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.

[32] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025.

[33] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, Mar 2025. Submitted on 31 Mar 2025 (v1).

A Appendix

A.1 Acknowledgement

We gratefully acknowledge the support of the Center for AI and Robotics (CAIR) at New York University Abu Dhabi for this research.

A.2 Experimental Setup

Model Configuration. All models used consistent sampling parameters: temperature=0.7, top-p=0.9, max tokens=8000. Models were loaded using vLLM [31] with 0.9 GPU memory utilization and eager execution for reproducibility.

Datasets. We evaluated on three datasets: GSM8K (1000 randomly sampled grade school math problems), MATH500 (500 problems across seven mathematical categories and five difficulty levels), and TriviaQA (1000 randomly sampled factual recall questions using RC no-context configuration).

Hardware. Experiments ran on a single NVIDIA A100 80GB GPU with 128GB system memory. Each complete layer ablation analysis required 4-11 hours depending on dataset size and model variant.

A.3 Detailed Methodology

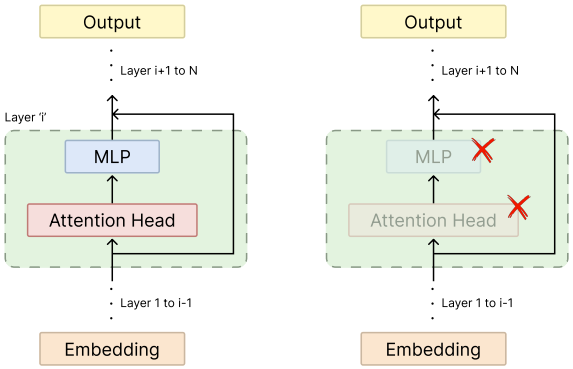

Figure 4: Zero ablation technique illustration. (a) Normal transformer layer with active MLP and attention. (b) Ablated layer with MLP and attention parameters set to zero, effectively nullified due to the skip connection

A.3.1 Zero-Ablation of Transformer Layers

As illustrated in Figure 5, our zero-ablation procedure systematically nullifies all parameters within a target layer while preserving the overall model architecture through its residual connections. For each target layer ℓ in a given model, we set the weight matrices and bias vectors in both the multi-head attention and MLP sublayers to zero. Specifically, for the multi-head attention mechanism, we ablate:

$$W_Q^{(\ell)}, W_K^{(\ell)}, W_V^{(\ell)}, W_O^{(\ell)} = \mathbf{0}$$
$$b_Q^{(\ell)}, b_K^{(\ell)}, b_V^{(\ell)}, b_O^{(\ell)} = \mathbf{0}$$

For the MLP component, we ablate:

$$W_{\text{gate}}^{(\ell)}, W_{\text{up}}^{(\ell)}, W_{\text{down}}^{(\ell)} = \mathbf{0}$$
$$b_{\text{gate}}^{(\ell)}, b_{\text{up}}^{(\ell)}, b_{\text{down}}^{(\ell)} = \mathbf{0}$$

Here, W_Q, W_K, W_V are the query, key, and value weight matrices; W_O is the output projection; and $W_{\text{gate}}, W_{\text{up}}, W_{\text{down}}$ are the gating, up-projection, and down-projection matrices in the MLP.

When the self-attention and MLP sub-layers are zeroed out, the operation of layer ℓ is reduced to its residual connections and normalization steps. Since we are looking at a pre-norm architecture, the layer becomes an identity layer:

$$\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)}$$

We conduct this analysis on two model families, Llama-3.1-8B [6] and Qwen-2.5-7B [7], examining four variants for each: (1) the pre-trained base models, (2) instruction-tuned models: Llama-3.1-8B-Instruct [6] and Qwen-2.5-7B-Instruct [7], (3) knowledge-distilled models: DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Qwen-7B [2], and (4) RLVR-trained models: Llama-3.1-8B-SimpleRL-Zoo [32] and Open-Reasoner-Zero-7B [33]. Our evaluation spans mathematical reasoning (GSM8K [8], MATH500 [9]) and factual recall (TriviaQA [10]) to distinguish task-specific processing patterns.

A.3.2 Representational Analysis via Normalized Mutual Information (NMI)

To understand *why* certain layers are critical, we analyze the evolution of the model's internal representations using Normalized Mutual Information (NMI) [26]. While prior work has qualitatively described how token representations cluster across transformer layers [22], our approach introduces a quantitative measure. We compute the NMI between the clustering of hidden states at layer 0 and the clustering of hidden states at each subsequent layer ℓ. This enables us to track how far each layer's cluster structure has diverged from the initial structure. See Appendix A.7 for more details on the prompts used.

Our procedure is as follows:

1. **Extract Hidden States:** For a given task input, we perform a single forward pass and store the hidden state representations for each layer. For each layer ℓ, we have a sequence of T token representations, $\mathbf{h}^{(\ell)} = (\mathbf{h}_1^{(\ell)}, \ldots, \mathbf{h}_T^{(\ell)})$, where each $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^d$ and d is the dimension of the hidden state.
2. **Establish Baseline Clustering:** We apply K-Means clustering to the T vectors from layer 0, $(\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_T^{(0)})$, to establish a set of baseline clusters, $\mathbf{C}^{(0)} = \{\mathbf{C}_1^{(0)}, \ldots, \mathbf{C}_K^{(0)}\}$. Note that we use zero-indexing, so layer 0 refers to the output of the first transformer layer.
3. **Cluster Subsequent Layers:** For each subsequent layer $\ell > 0$, we apply K-Means with the same number of clusters, K, to its hidden state vectors $(\mathbf{h}_1^{(\ell)}, \ldots, \mathbf{h}_T^{(\ell)})$ to obtain a new set of clusters, $\mathbf{C}^{(\ell)} = \{\mathbf{C}_1^{(\ell)}, \ldots, \mathbf{C}_K^{(\ell)}\}$.
4. **Calculate NMI:** We measure the similarity between the baseline clustering $\mathbf{C}^{(0)}$ and each subsequent layer's clustering $\mathbf{C}^{(\ell)}$. Given two clusterings $\mathbf{C}$ and $\mathbf{D}$, the NMI is formulated using the arithmetic mean for normalization: [26]

$$\text{NMI}(\mathbf{C}, \mathbf{D}) = \frac{I(\mathbf{C}, \mathbf{D})}{(H(\mathbf{C}) + H(\mathbf{D}))/2}$$

   This corresponds to the default method in the `sklearn.metrics.normalized_mutual_info_score` function used in our analysis. A detailed breakdown of each term—Mutual Information (I) and Entropy (H)—is provided in Appendix A.3.3.

A.3.3 Detailed NMI Calculation

This section provides a detailed breakdown of the terms used to calculate the Normalized Mutual Information (NMI) score, as referenced in subsubsection A.3.2. The calculation follows the

default implementation in the `scikit-learn` package. The final formula uses arithmetic mean normalization:

$$\text{NMI}(C, D) = \frac{I(C, D)}{(H(C) + H(D))/2}$$

The calculation begins by defining the properties of the clusterings. Let $C = \{c_1, \ldots, c_{|C|}\}$ and $D = \{d_1, \ldots, d_{|D|}\}$ be two different clusterings of the same N tokens. The probability of a token belonging to a specific cluster $c_i \in C$ is $P(i) = \frac{|c_i|}{N}$, and for a cluster $d_j \in D$ it is $P(j) = \frac{|d_j|}{N}$. The joint probability of a token belonging to both cluster c_i and cluster d_j is $P(i, j) = \frac{|c_i \cap d_j|}{N}$.

Using these probabilities, we first calculate the entropy of each clustering independently. The entropy $H(C)$ measures the uncertainty or disorder of a single clustering and is defined as:

$$H(C) = -\sum_{i=1}^{|C|} P(i) \log(P(i))$$

The entropy $H(D)$ is calculated identically for the second clustering.

Next, the mutual information, $I(C, D)$, measures the information shared between the two clusterings. It quantifies the reduction in uncertainty about one clustering given knowledge of the other. It is defined as:

$$I(C, D) = \sum_{i=1}^{|C|} \sum_{j=1}^{|D|} P(i, j) \log \left(\frac{P(i, j)}{P(i)P(j)} \right)$$

These components are then combined in the final formula to produce the normalized score, which provides a robust measure of similarity between cluster structures on a scale from 0 to 1. Note that the `scikit-learn` implementation uses the natural logarithm.

A.4 Prompt Templates

We used model-specific prompts optimized for each architecture family.

Mathematical Reasoning (GSM8K & Math500):

Qwen Models:

```
<|im_start|>system
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>user
{problem}
<|im_end|>
<|im_start|>assistant
```

Llama Base:

```
Question: {problem}
Answer: Let's think step by step.
```

Llama Instruct:

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are a helpful mathematics assistant. Please solve the problem step by
step and provide your final answer within \boxed{}.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{problem}<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

DeepSeek Distilled Models:

```
A conversation between User and Assistant. The User asks a question, and the
Assistant solves it. The Assistant first thinks about the reasoning process
in the mind and then provides the User with the answer. The reasoning process
```

```
is enclosed within <think> </think> and answer is enclosed within
<answer> </answer> tags, respectively.
User: {problem}
Assistant: <think>
```

Llama RL Model:

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{problem}
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>assistant
```

Factual Recall (TriviaQA):

Qwen Models:

```
<|im_start|>system
Please answer the trivia question directly and concisely.<|im_end|>
<|im_start|>user
{question}
<|im_end|>
<|im_start|>assistant
```

Llama Base & RL:

```
Question: {question}
Answer: Let's think step by step.
```

Llama Instruct:

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are a helpful assistant. Please answer the trivia question directly
and accurately.<|eot_id|>
<|start_header_id|>user<|end_header_id|>
{question}<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

DeepSeek Distilled Models:

```
A conversation between User and Assistant. The User asks a question, and the
Assistant solves it. The Assistant first thinks about the reasoning process
in the mind and then provides the User with the answer. The reasoning process
is enclosed within <think> </think> and answer is enclosed within
<answer> </answer> tags, respectively.
User: {question}
Assistant: <think>
```

A.5 Answer Extraction and Evaluation

Mathematical Problems. We used the `math_verify` library to parse numerical answers from model outputs, handling various formats including `\boxed{}` notation, explicit numerical statements, and embedded calculations.

Factual Recall. For TriviaQA, we employed case-insensitive substring matching, checking for ground truth answers (including aliases) within model responses.

A.6 Sample Responses by Important Layers:

Below are the sample responses of base model and layer ablated models. Specifically, we show the responses for the original base model and the model with most significant layer ablated.

Sample Question:

```
Joy can read 8 pages of a book in 20 minutes. How many hours will it take her to
↪  read
120 pages?
```

Qwen Base Original (Correct):

```
Joy can read 8 pages in 20 minutes, so she can read 8/20 = 0.4 pages per minute.
To read 120 pages, it will take her 120/0.4 = 300 minutes.
There are 60 minutes in an hour, so it will take her 300/60 = 5 hours.
The answer is 5.
```

Qwen Base Layer 23 Ablated (Incorrect):

```
To determine how many hours it will take Joy to read 120 pages, we can follow
↪  these
steps:
1. Find the number of pages Joy reads per minute: Joy reads 8 pages in 20
↪  minutes, so
she reads 8/20 = 0.4 pages per minute.
2. Determine the number of minutes it takes to read 120 pages: Since Joy reads
↪  0.4
pages per minute, it will take her 120/0.4 = 2400 minutes to read 120 pages.
3. Convert the minutes to hours: Since 1 hour is equal to 60 minutes, we can
↪  convert
2400 minutes to hours by dividing by 60: 2400/60 = 120/3 = 60/2 = 30 hours.

So, it will take Joy 30 hours to read 120 pages.
```

Llama Base Original (Correct):

```
Joy can read 8 pages in 20 minutes. So, she can read 1 page in 20/8 minutes. That
↪  is,
she can read 1 page in 2.5 minutes. So, she can read 120 pages in 120 x 2.5
↪  minutes.
That is, she can read 120 pages in 300 minutes. That is, she can read 120 pages
↪  in 5
hours.
```

Llama Base Layer 15 Ablated (Incorrect):

```
Joy can read 8 pages in 20 minutes. So, she can read 1 page in 20/8 minutes. That
means she can read 1 page in 4 minutes. Now, we know that 1 page takes 4 minutes.
So, 120 pages will take 120 x 4 = 480 minutes. 480 minutes is 480/60 = 96 hours.
So, Joy will take 96 hours to read 120 pages.
```

A.7 NMI Prompts

We perform NMI analysis using a single prompt that concatenates 20 math problems from the MATH500 dataset, each followed by its solution. Note that each of these 20 problems were sampled randomly for each run and for each model family. The NMI score were averaged across 5 runs. These problems were selected by taking two from each of the seven categories in the dataset to ensure broad coverage. The prompt is used across all model variants and passed through the model once - without any decoding - to extract hidden states from each layer. The full procedure is described in subsubsection A.3.2. A sample excerpt showing 2 of one of the 20 sampled problems is shown below.

Similarly, we perform NMI analysis on the TriviaQA task using a single prompt that concatenates 20 random question-answer pairs. As above, the prompt is passed through the model once - without any

decoding - to extract hidden states across layers. A sample excerpt showing 5 of the 20 questions is shown below.

Our main goal for this analysis was to see how the model organizes its representations when given a prompt with a wide variety of tokens and concepts. To achieve this diversity, we concatenated multiple question-and-solution pairs into a single prompt. We used a single forward pass on this long prompt because of the unique challenge it presents. While the self-attention mechanism allows every token to see every other token, the need to form a coherent representation for each distinct problem creates a strong incentive for the model to learn to differentiate between their contexts. This is a more demanding task than processing a simple, isolated prompt, and it gives us a better view of how the model uses its representations to manage and insulate different contexts. While ideally, we might use a single, extremely long problem, our approach is a practical way to achieve the high diversity needed for this analysis.

Sample Prompt for NMI Analysis on MATH task:

Question: Convert the point $(0, 3)$ in rectangular coordinates to polar coordinates. Enter your answer in the form (r, θ), where $r > 0$ and $0 \le \theta < 2\pi$.

Answer: We have that $r = \sqrt{0^2 + 3^2} = 3$. Also, if we draw the line connecting the origin and $(0, 3)$, this line makes an angle of $\frac{\pi}{2}$ with the positive x-axis. Therefore, the polar coordinates are $\boxed{\left(3, \frac{\pi}{2}\right)}$.

Question: The set of points (x, y, z) that satisfy

$$2x = 3y = -z$$

is a line. The set of points (x, y, z) that satisfy

$$6x = -y = -4z$$

is another line. Find the angle between these lines, in degrees.

Answer: For the first line, let $t = 2x = 3y = -z$. Then

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} t/2 \\ t/3 \\ -t \end{pmatrix} = \frac{t}{6} \begin{pmatrix} 3 \\ 2 \\ -6 \end{pmatrix}.$$

Thus, the direction vector of the first line is $\begin{pmatrix} 3 \\ 2 \\ -6 \end{pmatrix}$. For the second line, let $t = 6x = -y = -4z$. Then

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} t/6 \\ -t \\ -t/4 \end{pmatrix} = \frac{t}{12} \begin{pmatrix} 2 \\ -12 \\ -3 \end{pmatrix}.$$

Thus, the direction vector of the second line is $\begin{pmatrix} 2 \\ -12 \\ -3 \end{pmatrix}$. Note that

$$\begin{pmatrix} 3 \\ 2 \\ -6 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -12 \\ -3 \end{pmatrix} = 0.$$

Hence, the angle between the lines is $\boxed{90^\circ}$. ...

Sample Prompt for NMI Analysis on TriviaQA:

Question: How old was Jimi Hendrix when he died?
Answer: 27

Question: Who was the next British Prime Minister after Arthur Balfour?
Answer: henry campbell bannerman

Question: In what year's Olympics were electric timing devices and a public-address system used for the first time?
Answer: in 1912 in stockholm

Question: Where did the Shinning Path terrorists operate?
Answer: republic of perú

Question: What was the last US state to reintroduce alcohol after prohibition?
Answer: history of mining in utah ...

A.8 Results on Qwen Models of Various Sizes

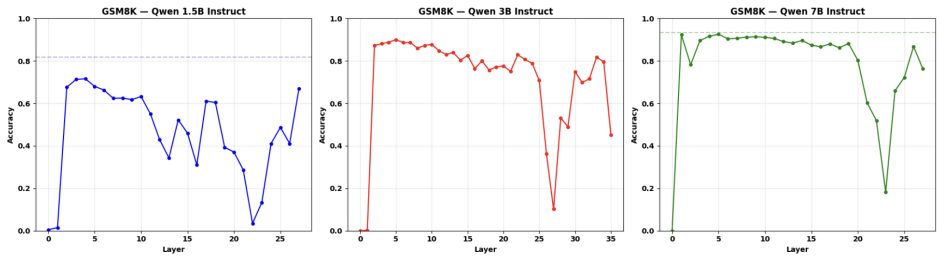

Figure 5: Zero ablation on Qwen2.5-(1.5B,3B,7B)-Instruct Models. We find that all 3 models have critical layers at relatively similar positions.

A.9 Testing Logical and Airthmetic Errors

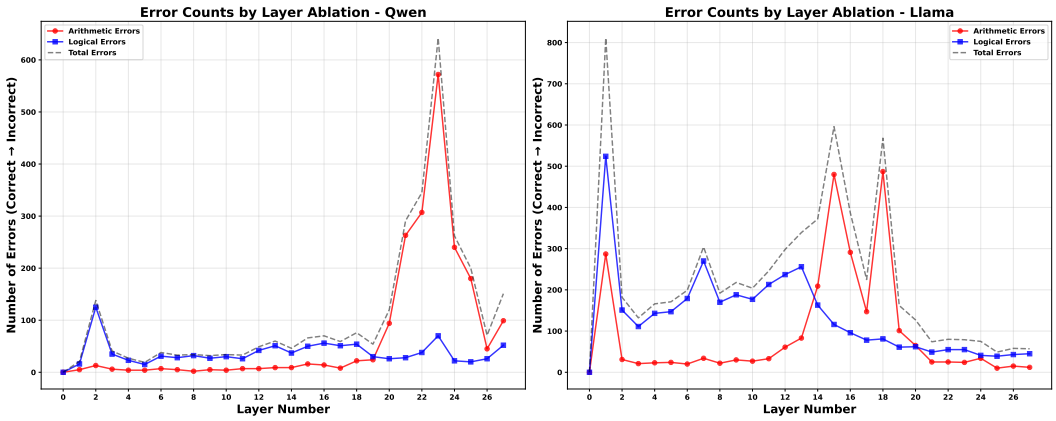

Figure 6: Error counts by layer ablation for Qwen (left) and Llama (right) instruct models. We used the GPT API to automatically classify model responses into arithmetic errors (red) and logical errors (blue). Notably, for Qwen, layer 23 shows a strong spike in arithmetic errors despite logically correct reasoning, suggesting that arithmetic mistakes dominate at critical layers. In contrast, logical errors remain more evenly distributed across layers. For the Qwen base model (not shown here), early layers (e.g., layer 2) produced repetitive and incoherent responses

A.10 TriviaQA NMI Plot

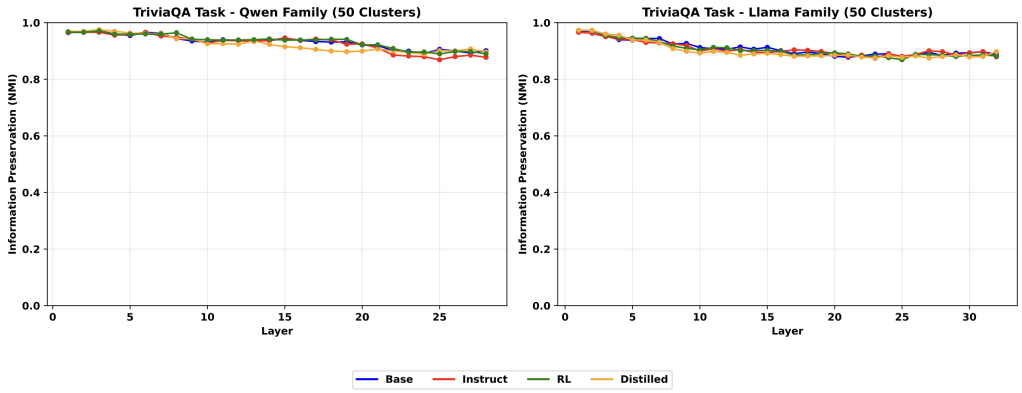

Figure 7: NMI remains relatively stable for triviaQA

A.11 Residual Norm analysis

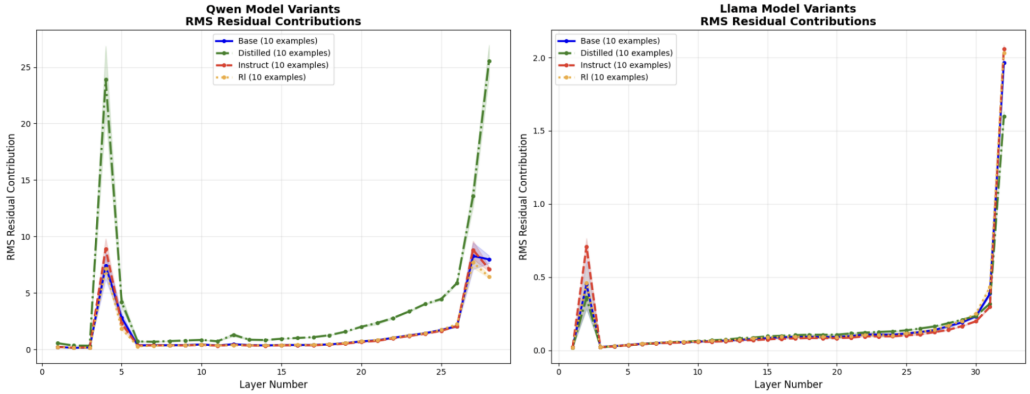

Figure 8: Root-mean-square (RMS) norm of residual stream contributions by layer for Qwen (left) and LLaMA (right) model variants. The curves show how much each layer adds to the residual stream, with largely consistent patterns across Base, Distilled, Instruct, and RL variants.