What Affects the Effective Depth of Large Language Models?

Yi Hu[1] **Cai Zhou**[2] **Muhan Zhang**[1,†]
[1]Institute for Artificial Intelligence, Peking University
[2]Department of Computer Science, Massachusetts Institute of Technology

Abstract

The scaling of large language models (LLMs) emphasizes increasing depth, yet performance gains diminish with added layers. Prior work introduces the concept of "effective depth", arguing that deeper models fail to fully utilize their layers for meaningful computation. Building on this, we systematically study how effective depth varies with model scale, training type, and task difficulty. First, we analyze the model behavior of Qwen-2.5 family (1.5B32B) and find that while the number of effective layers grows with model size, the effective depth ratio remains stable. Besides, comparisons between base and corresponding long-CoT models show no increase in effective depth, suggesting that improved reasoning stems from longer context rather than deeper per-token computation. Furthermore, evaluations across tasks of varying difficulty indicate that models do not dynamically use more layers for harder problems. Our results suggest that current LLMs underuse available depth across scales, training paradigms and tasks of varying difficulties, pointing out research opportunities on increasing the layer utilization rate of LLMs, model pruning, and early exiting. Our code is released at `https://github.com/AheadOFpotato/what_affects_effective_depth`.

1 Introduction

The scaling of large language models (LLMs) [1, 2, 3, 4, 5] has consistently emphasized increased depth, with empirical evidence suggesting that model performance improves with additional layers—despite diminishing returns. As pointed out by Csordás et al. [6], this trend raises a fundamental question: are these models truly leveraging their depth to perform more complex, hierarchical computations, or are they merely distributing similar computational operations over a greater number of layers?

Csordás et al. [6] reveals a striking under-utilization of depth: layers in the second half are simply refining existing representations rather than contributing to novel feature composition or conducting deeper reasoning. The study introduces the concept of "effective depth" and suggests that inefficient depth utilization may be a fundamental cause of diminishing scaling returns. Building directly upon this foundation, our work seeks to systematically investigate the factors that influence this effective depth. We aim to achieve a more comprehensive understanding of how depth utilization behaves across model scale, specialized training, and task difficulty. Our findings are as follows:

1. **Regarding model size.** Following the methodologies established in prior work, we first analyze the Qwen-2.5 model family (from 1.5B to 32B) [2] using a suite of techniques including residual cosine similarity, logit lens, layer effects on future computations, residual erasure and integrated gradients [6, 7]. Our results confirm the core phenomenon: there exists a phase transition where early layers drive feature composition and later layers engage in minor refinements. Furthermore, while the absolute number of these "effective" layers increases with model size, the ratio of

effective depth to total depth remains stable. This aligns with the conclusions of Csordás et al. [6] that larger models do not fundamentally alter their computational strategy; they simply replicate the same utilization pattern over a larger number of layers, rather than using the extra depth to invent new types of computation. This finding provides a nuanced explanation for diminishing returnswider models gain new capabilities, while deeper models primarily gain precision.

2. **Regarding long-CoT models.** Given that long-CoT models have demonstrated exceptional performance in complex reasoning tasks [8, 9], a natural hypothesis is that they might achieve this by more effectively exploiting their depth for "deeper" reasoning in each forward pass. To test this, we compare the effective depth of base and instruct models in the Qwen-2.5 model family [2] against their corresponding DeepSeek-R1-distill counterparts [8]. Surprisingly, our analysis reveals no significant increase in effective depth. The enhanced reasoning performance appears not to be driven by a fundamental change in how the model utilizes its layers during each forward pass. Instead, the gains are likely attributable to the model's optimized ability to reason over longer sequences, not to deeper computation within a single token's forward process.
3. **Regarding task difficulty.** We further probe whether models dynamically allocate their depth based on computational demand. One might expect harder problems to require and therefore activate deeper layers. We evaluate models on a difficulty spectrum from HellaSwag (natural language understanding) [10] to GSM8K (grade school math) [11] to AIME24 (high school math contests) [12]. Counter to intuition, the effective depth remains largely consistent across all tasks. The model does not appear to leverage significantly more of its depth for harder problems.

In summary, modern LLMs, across scales, specialized training regimes and task difficulties, fail to fully exploit their available depth for composing novel, high-level features.

2 Preliminary

We mainly focus on the Qwen-2.5 model family [2] (including base models and instruct models), and their corresponding DeepSeek-R1-Distill versions [8]. They are all pre-norm Transformers [13, 14] and the forward process of a layer l is as follows:

$$\boldsymbol{a}_l = \mathrm{SelfAttention}_l(\mathrm{RMSNorm}(\boldsymbol{h}_l)) \tag{1}$$

$$\hat{\boldsymbol{h}}_l = \boldsymbol{h}_l + \boldsymbol{a}_l \tag{2}$$

$$\boldsymbol{m}_l = \mathrm{MLP}_l(\mathrm{RMSNorm}(\hat{\boldsymbol{h}}_l)) \tag{3}$$

$$\boldsymbol{h}_{l+1} = \hat{\boldsymbol{h}}_l + \boldsymbol{m}_l \tag{4}$$

Here, $\boldsymbol{h}_l \in \mathbb{R}^{n_{\text{context}} \times d_{\text{model}}}$ is the residual stream [15], $\boldsymbol{a}_l, \boldsymbol{m}_l$ are the outputs of the SelfAttention layers and MLP layers, which are directly added back to the residual stream. n_{context} is the length of the input sequence, and d_{model} is the dimension of the hidden states of the model. RMSNorm [16] is adopted in the Qwen-2.5 model family to replace traditional layer normalization [13]. Following Csordás et al. [6], we denote $\mathrm{SelfAttention}_l(\cdot)$ and $\mathrm{MLP}_l(\cdot)$ as "sublayers".

The residual stream starts with $\boldsymbol{h}_0 = \mathrm{Embedding}(x)$, where $x \in \mathbb{N}^{n_{\text{context}}}$ is the sequence of `token_ids`. Then the final results of residual stream goes through the output layer and results in the output probability distribution over vocabulary: $\boldsymbol{y} = \mathrm{softmax}(\mathrm{RMSNorm}(\boldsymbol{h}_L)\boldsymbol{W}^{out})$, where $\boldsymbol{y} \in \mathbb{R}^{n_{\text{context}} \times |V|}$, $\boldsymbol{W}^{out} \in \mathbb{R}^{d_{\text{model}} \times |V|}$, L is the number of layers in the model, V is the vocabulary.

3 Methods

Csordás et al. [6] proposes a suite of methods to qualitatively probe the effective depth. We introduce and extend the methods to qualitatively assess effective depth across different models and datasets:

Residual cosine similarity. Residual cosine similarity measures how each layer or sublayer interacts with the residual stream. For a given layer l, we compute the cosine similarity between its contribution (the output of either SelfAttention $\boldsymbol{a}_l$, MLP $\boldsymbol{m}_l$, or their sum) and the resulting residual state $\boldsymbol{h}_l$. Formally, the similarities are defined as $\mathrm{cosim}(\boldsymbol{a}_l + \boldsymbol{m}_l, \boldsymbol{h}_l)$ for the full layer, $\mathrm{cosim}(\boldsymbol{a}_l, \boldsymbol{h}_l)$ for self-attention, and $\mathrm{cosim}(\boldsymbol{m}_l, \boldsymbol{h}_l + \boldsymbol{a}_l)$ for the MLP. The intuition is that a cosine similarity near zero suggests the module writes a new, orthogonal feature into the residual stream; negative values indicate feature erasure; and positive values signify the amplification of an existing feature.

Table 1: Effective depth (ED) and effective depth ratio (ratio $= \frac{\text{ED}+1}{\text{L}}$) across base, instruct, and long-CoT models of different sizes (1.5B to 32B parameters) and on datasets with varying difficulty.

	Cosine Similarity						**Logit Lens KL**						**Logit Lens Overlap**					
	HellaSwag		GSM8K		AIME24		HellaSwag		GSM8K		AIME24		HellaSwag		GSM8K		AIME24	
	ED	ratio	ED	ratio	ED	ratio	ED	ratio	ED	ratio	ED	ratio	ED	ratio	ED	ratio	ED	ratio
DS-R1-Qwen-1.5B	17	0.64	16	0.61	17	0.64	20	0.75	1	0.07	24	0.89	23	0.86	23	0.86	24	0.89
Qwen2.5-1.5B-Instruct	16	0.61	20	0.75	19	0.71	21	0.79	22	0.82	23	0.86	23	0.86	23	0.86	23	0.86
Qwen2.5-Math-1.5B	16	0.61	16	0.61	16	0.61	20	0.75	22	0.82	23	0.86	23	0.86	23	0.86	23	0.86
DS-R1-Qwen-7B	16	0.61	16	0.61	16	0.61	24	0.89	24	0.89	24	0.89	25	0.93	25	0.93	24	0.89
Qwen2.5-7B-Instruct	17	0.64	20	0.75	18	0.68	25	0.93	25	0.93	25	0.93	26	0.96	26	0.96	26	0.96
Qwen2.5-Math-7B	16	0.61	11	0.43	16	0.61	23	0.86	23	0.86	23	0.86	24	0.89	24	0.89	24	0.89
DS-R1-Qwen-14B	26	0.56	30	0.65	30	0.65	40	0.85	39	0.83	41	0.88	44	0.94	44	0.94	44	0.94
Qwen2.5-14B-Instruct	27	0.58	32	0.69	30	0.65	40	0.85	41	0.88	42	0.90	45	0.96	45	0.96	45	0.96
Qwen2.5-14B	27	0.58	30	0.65	30	0.65	40	0.85	40	0.85	42	0.90	45	0.96	45	0.96	45	0.96
DS-R1-Qwen-32B	42	0.67	42	0.67	46	0.73	58	0.92	55	0.88	57	0.91	61	0.97	58	0.92	58	0.92
Qwen2.5-32B-Instruct	43	0.69	46	0.73	43	0.69	60	0.95	58	0.92	58	0.92	61	0.97	60	0.95	60	0.95
Qwen2.5-32B	43	0.69	46	0.73	46	0.73	60	0.95	57	0.91	59	0.94	61	0.97	59	0.94	60	0.95

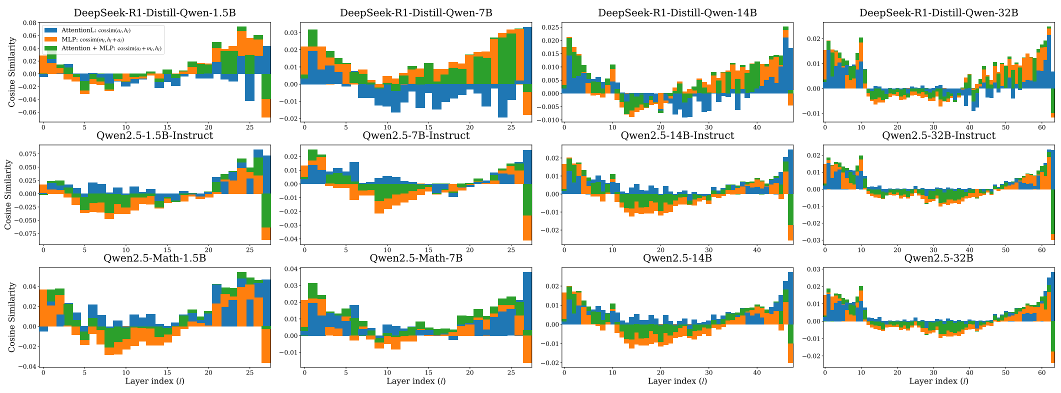


Figure 1: Cosine similarity of (sub)layer contributions and the residual evaluated on GSM8K.

Logit Lens. Logit lens evaluates how early the models output distribution begins to stabilize. We decode the hidden state $\boldsymbol{h}_l$ using the models output projection and compute the KL divergence between this early distribution and the models final distribution. Additionally, we measure the overlap between the top-5 tokens from this intermediate distribution and from the final distribution.

Layer effects on future computation. Here we probe the influence of skipping a layer on subsequent computations. For a given prompt, we first run a forward pass to record the residual states h_l. We then intervene by skipping a specific layer s for all token positions $t \leq t_s$ (where t_s is a sampled position within the sequence), effectively setting $\bar{h}_{s+1} := \bar{h}_s$ for those tokens. The effect of this intervention is measured on the subsequent tokens ($t > t_s$) by computing the relative change in the contribution of a later layer $l > s$: $\|(h_{l+1} - h_l) - (\bar{h}_{l+1} - \bar{h}_l)\|_2 / \|h_{l+1} - h_l\|_2$. The maximum value of this metric across multiple prompts and sequence positions is taken. We also compare the final output probabilities directly via $\|y - \bar{y}\|_2$.

Residual erasure. Residual erasure identifies until which layer information from a specific token remains relevant for the final prediction. For a token at position t and layer l, we intervene by replacing its residual vector $h_{l+1}[t]$ with an uninformative baselinethe average residual vector at layer l computed over a dataset (GSM8K here), while leaving all other tokens unchanged. The effect is quantified as the maximum change in prediction norm ($\|y - \bar{y}\|_2$) among all answer tokens.

Integrated gradients. The metric attributes the models prediction on the answer tokens to contributions from each layer. We compute the gradients of the output logits for all answer tokens with respect to the activation at each layer and each token position.

Beyond these qualitative probes, we introduce two quantitative measures to compare effective depth across models and datasets. For *residual cosine similarity*, we average the similarity scores across layers, MLPs, and SelfAttention modules, and identify the effective depth as the point where the averaged similarity transitions from negative to positive. For the *logit lens*, we use two metrics: we define the effective depth as the layer where the KL divergence from the final output drops below half of its maximum observed value, and alternatively, as the layer where the top-5 token overlap with the final output first exceeds 0.3.

4 Experiments

4.1 Does Model Size Affect Effective Depth?

The residual cosine similarity, shown in Figure 1, exhibits a consistent pattern across models: an initial positive phase, followed by a decline into negative values, and a final return to positive. The initial near-zero similarity in shallow layers suggests context integration, while the subsequent positive phase corresponds to feature refinement. The first half of the network is predominantly characterized by feature erasure (negative similarity), until a sharp phase transition occurs near the middle layers, after which the model begins strengthening existing features.

We quantify the corresponding depth of this transition in Table 1 (Cosine Similarity). The results show that the effective depth ratio remains remarkably stable. This indicates that larger models contain a growing number of "ineffective" layers that do not contribute to feature composition.

The logit lens analysis, as shown in Figure 2, further supports this conclusion. The KL divergence between intermediate and final predictions shows a sharp drop in the second half of the network, while the top-5 token overlap exhibits a concurrent sharp rise. Together, these indicate a transition from computation to refinement. As quantified in Table 1, the depth of this transition, measured both by KL divergence (half-max point) and overlap (exceeding 0.3), is slightly less consistent across scales than the cosine similarity metric, with a mild increasing trend in ratio for larger models.

Furthermore, the effect of skipping layers on downstream computations, illustrated in Figure 3, reveals that layers in the second half have substantially less influence on both later layers and final output predictions. This pattern is consistent across all model sizes, with similar decay profiles.

Finally, results from integrated gradients (Figure 4(a)) and residual erasure (Figure 4(b)) show that the dependence of answer token predictions on earlier layers declines markedly in the second half of the network. The position of this decline remains stable relative to network depth across model sizes.

4.2 Do Long-CoT Models Think Deeper?

Given that long-CoT models demonstrate superior performance on complex reasoning [8, 9], one might hypothesize that they achieve this by utilizing deeper computations within each forward pass. To test this, we compare the effective depth of DeepSeek-R1-Distill models [8] against their corresponding base models [2]. As summarized in Table 1, we find no significant difference in effective depth ratio between long-CoT and base models. This consistency is further illustrated across all probing methods: residual cosine similarity (Figure 1), logit lens (Figure 2), layer-skipping effects (Figure 3), integrated gradients (Figure 4(a)), and residual erasure (Figure 4(b)). The results are consistent that long-CoT models do not exhibit a deeper utilization of the network.

4.3 Does Task Difficulty Affect Effective Depth?

We next investigate whether models dynamically adjust their effective depth in response to computational demand, expecting that harder tasks might engage deeper layers. We evaluate models on three tasks of increasing difficulty: HellaSwag (natural language understanding) [10], GSM8K (grade school math) [11], and AIME24 (high school math contests) [12]. Results in Table 1 show that effective depth remains largely consistent across all tasks, indicating that model depth utilization is not adaptive to problem difficulty. Additional results are provided in Appendix D, including residual cosine similarity (Figure 5), effects of skipping layers on future computations (Figure 6) and output distributions (Figure 7), as well as logit lens KL divergence (Figure 8) and token overlap (Figure 9).

5 Conclusion

In this work, we provide a comprehensive study of the factors that may affect effective depth in LLMs, including model scales, training strategies, and task difficulties. First, the effective depth ratio remains roughly constant with the increase of model size. Second, long-CoT models show no increase in effective depth despite their enhanced reasoning capabilities. Third, effective depth remains consistent across task difficulty, indicating no dynamic depth allocation based on computational demand. These results demonstrate that LLMs fail to fully exploit their architectural depth.

References

[1] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[2] Qwen Team. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[3] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[4] DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

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

[6] Róbert Csordás, Christopher D Manning, and Christopher Potts. Do language models use their depth efficiently? *arXiv preprint arXiv:2505.13898*, 2025.

[7] Nostalgebraist. Interpreting gpt: The logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[8] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[9] OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

[10] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[11] 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.

[12] MAA. American invitational mathematics examination-aime 2024, 2024. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

[13] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[16] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Appendices

A Limitations

This work follows the methodology of Csordás et al. [6] to comprehensively analyze factors influencing effective depth. We introduce quantitative metrics based on residual cosine similarity and logit lens to compare effective depth across models and datasets. However, the proposed metrics—particularly the two variants of logit lens—remain relatively straightforward and exhibit some instability. Developing more robust and well-validated measures of effective depth is an important direction for future research.

Furthermore, while we confirm and extensively analyze the phenomenon of depth under-utilization across model scales, training strategies, and task demands, this study does not propose solutions to improve layer utilization. Our findings highlight the need for future work to explore architectural or training approaches that enable models to leverage their full depth more effectively.

B Model Details

Our analysis focuses on the Qwen-2.5 model family [2]. For base models, we use the same versions selected by DeepSeek-AI [8]: Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, Qwen2.5-14B, and Qwen2.5-32B. For instruction-tuned models, we use the standard instruct variants from the Qwen-2.5 family: Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, and Qwen2.5-32B-Instruct. Additionally, we include the corresponding DeepSeek-R1-Distill versions derived from these base models: DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-14B, and DeepSeek-R1-Distill-Qwen-32B.

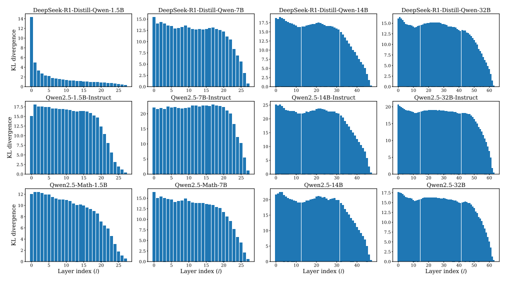
Models of the same size share identical architectures. The architectural details are provided in Table 2.

Table 2: Model details.

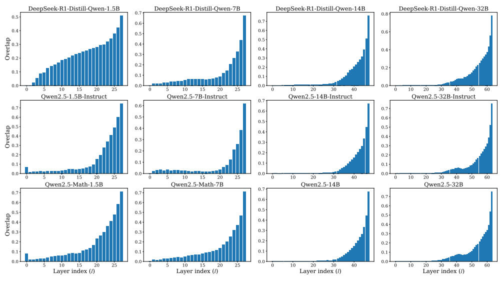
Models	Layers	Heads (Q/KV)
1.5B	28	12 / 2
7B	28	28 / 4
14B	48	40 / 8
32B	64	40 / 8

C Additional Effective Depth Results on GSM8K

We show the results of logit lens in Figure 2, the effects of skipping a layer on future computations in Figure 3(a) and on output distributions in Figure 3(b). Besides, the results of integrated gradients residual erasure are shown in Figure 4(a) and Figure 4(b) respectively.

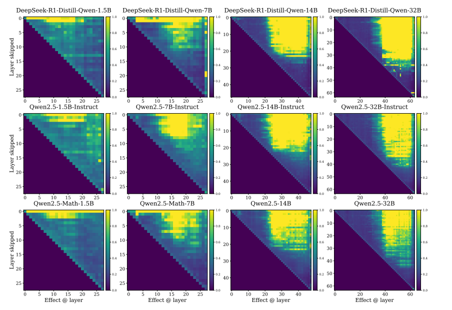


(a) Logit lens KL Divergence.

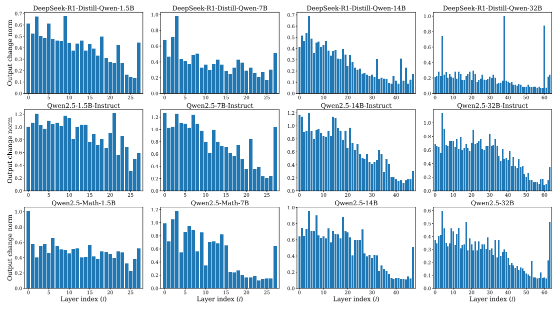


(b) Logit lens Overlap.

Figure 2: Logit lens Results on GSM8K.

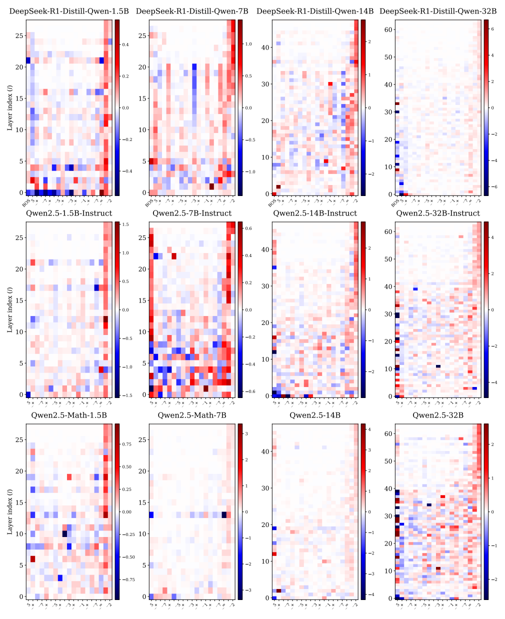


(a) Effects of skipping a layer on later layers' contribution.

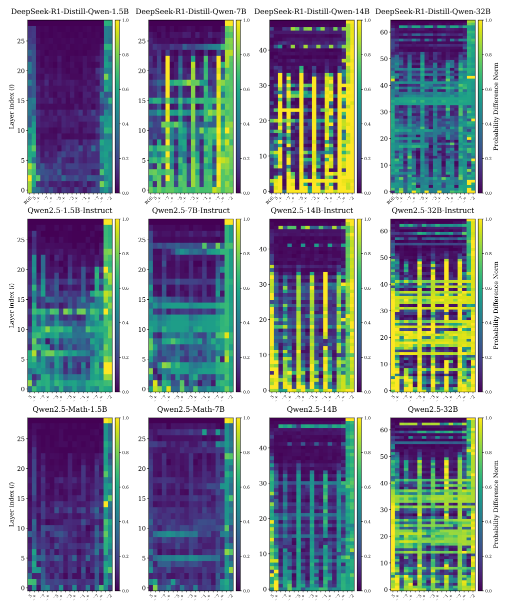


(b) Effects of skipping a layer on output norm.

Figure 3: Effect of skipping a layer on future computation evaluated on GSM8K.



(a) Integrated gradients on addition.



(b) Residual erasure on addition.

Figure 4: The Effects of individual computation steps evlauated on GSM8K.

D Effective Depth of All Models Evaluated on GSM8K and HellaSwag

We show the results of effective depth of Qwen-2.5 family (base and instruct models) and their long-CoT variants tested on GSM8K and HellaSwag, including residual cosine similarity results in Figure 5; the effects of skipping a layer on future computations in Figure 6 and on output distributions in Figure 7; logit lens KL divergence in Figure 8; logit lens overlap in Figure 9.

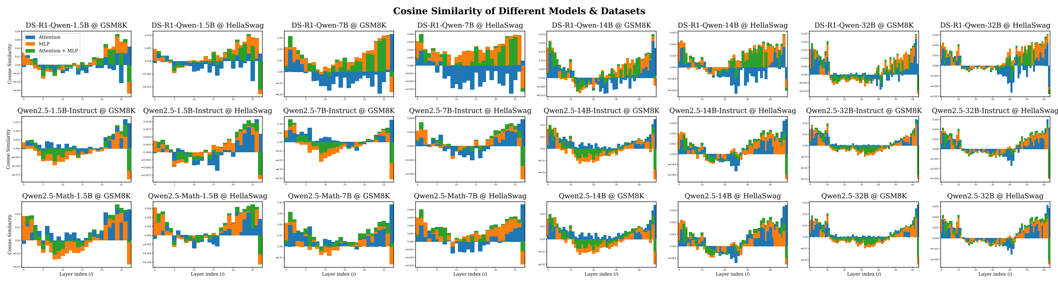


Figure 5: Residual cosine similarity of all models on GSM8K and HellaSwag.

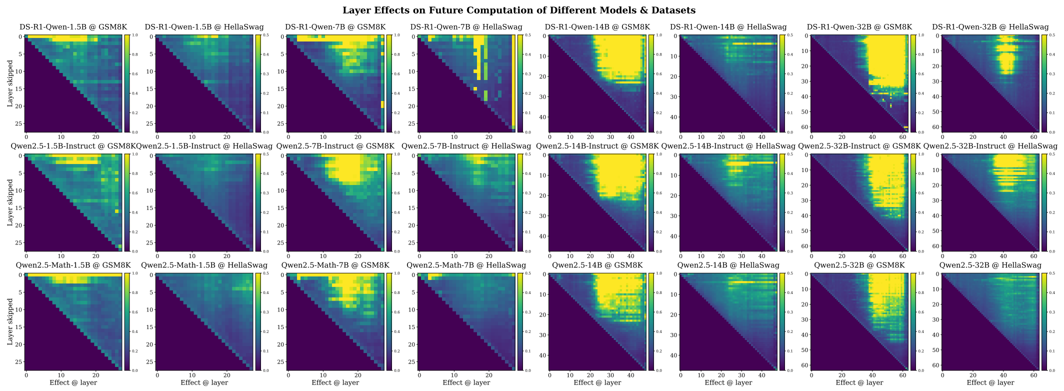


Figure 6: The effects of skipping a layer on future computations, the results include all models on GSM8K and HellaSwag.

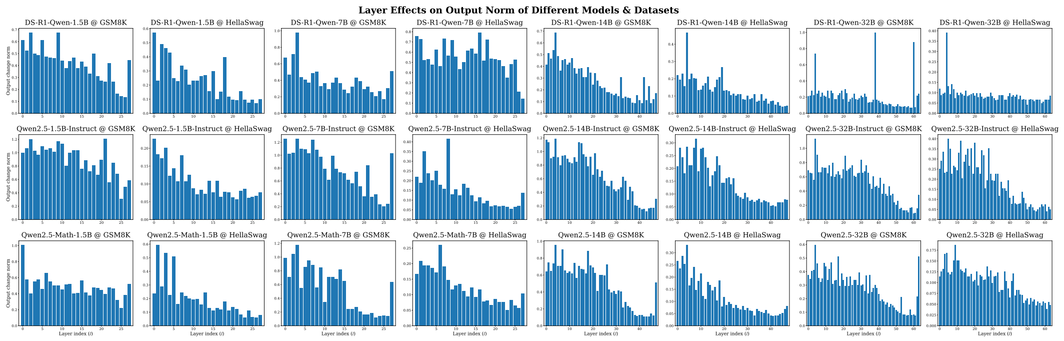


Figure 7: The effects of skipping a layer on output distributions, the results include all models on GSM8K and HellaSwag.

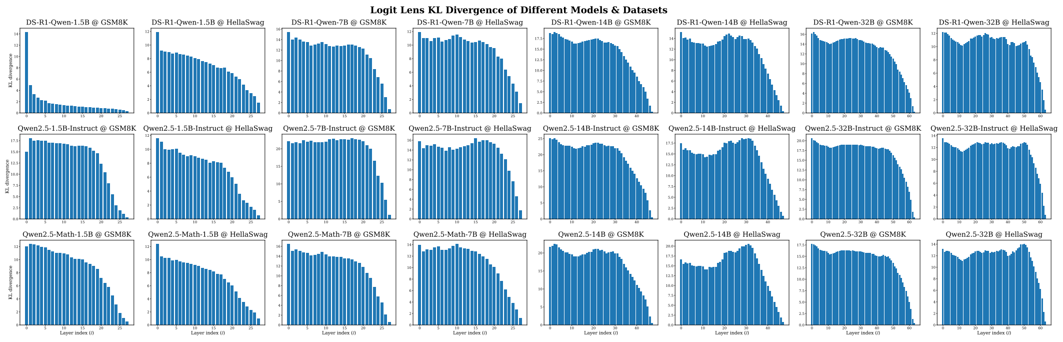


Figure 8: Logit lens KL divergence between early layer distributions and the final distributions. The results include all models on GSM8K and HellaSwag.

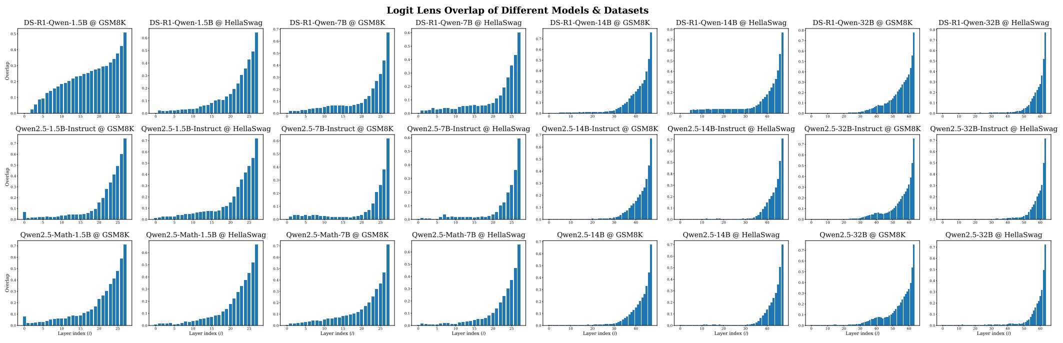


Figure 9: Logit lens top-5 overlap between early layer distributions and the final distributions. The results include all models on GSM8K and HellaSwag.