From Parameters to Performance: A Data-Driven Study on LLM Structure and Development

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved remarkable success across various domains, driving significant technological advancements and innovations. Despite the rapid growth in model scale and capability, systematic, data-driven research on how structural configurations affect performance remains scarce. To address this gap, we present a large-scale dataset encompassing diverse open-source LLM structures and their performance across multiple benchmarks. Leveraging this dataset, we conduct a systematic, data mining-driven analysis to uncover the relationship between structural configurations and performance. Our study begins with a review of the historical development of LLMs and an exploration of potential future trends. We then analyze how various structural choices impact performance across benchmarks and validate our findings using mechanistic interpretability techniques. By providing data-driven insights into LLM optimization, our work aims to guide the targeted development and application of future models.

1 Introduction

Large language models (LLMs) have revolutionized a wide range of domains, including natural language understanding and generation (Radford et al., 2019), as well as multimodal applications (Achiam et al., 2023), driving significant advancements in both technology and real-world applications. Models such as GPT-3 (Brown et al., 2020), Qwen (Bai et al., 2023), and Llama (Touvron et al., 2023a) have demonstrated outstanding performance by leveraging scaling laws (Kaplan et al., 2020), which link improvements in model performance with increases in model size, training data, and computational resources. These models have set new benchmarks across various fields. However, despite the remarkable progress in scaling up these models, a systematic exploration of the relationship between structural configurations and task-specific performance remains lacking.

As LLMs become increasingly complex and resource-intensive, deploying these models in real-world applications presents significant challenges in terms of cost and energy consumption (Zhao et al., 2023; Kaddour et al., 2023). While structural configurations are known to influence model performance (Yang et al., 2024b; Dong et al., 2023), their effects across different tasks and application domains have not been comprehensively analyzed. The growing complexity of LLMs necessitates a deeper exploration of the trade-offs between various structural designs, computational resources, and model performance.

To address these challenges, we present a large-scale dataset encompassing various open-source LLMs structural configurations and their performance across multiple benchmarks, providing a foundation for data-driven insights into the relationship between model structure and performance. This paper reviews the historical development of LLMs and explores how structural configurations impact LLMs performance. Additionally, we employ mechanistic interpretability techniques to investigate the mechanism of models across diverse benchmarks, further validating the phenomena uncovered in the dataset. Through this analysis, we provide valuable insights for optimizing LLMs design, contributing to the development of models that are not only powerful and scalable but also efficient and adaptable to diverse applications.

Our key contributions are summarized as follows:

- **Large-Scale Open-Source LLMs Structure and Performance Dataset:** We introduce a large-scale dataset containing a variety of open-source LLMs structural configurations and their performance on multiple benchmarks, offering a foundation for data-driven

insights into the relationship between model structure and performance.

- **Study on the Impact of Structure on Performance:** We systematically examine the influence of structural configurations on LLMs performance, focusing on key parameters such as layer depth.

- **Mechanistic Interpretability Analysis and Validation:** We employ layer-pruning and gradient analysis techniques to validate the findings regarding the impact of layer depth on performance across different benchmarks, as mined from the LLMs structure and performance dataset.

2 Related Work

2.1 Model Evaluation

In the field of LLMs, evaluating and comparing model performance is crucial for advancing technology. One of the most prominent platforms for benchmarking is the Open LLM Leaderboard (*the leaderboard*, Beeching et al., 2023; Fourrier et al., 2024), hosted by HuggingFace, which provides a standardized environment for evaluating various large-scale models across numerous tasks.

Although *the leaderboard* provides practical performance comparisons between LLMs, it overlooks the structural configurations of the models. There has been limited exploration of the relationships between these configurations and the performance across different datasets. Our work aims to address this gap by combining model structural configurations with performance data from *the leaderboard*. This additional dimension provides valuable insights into how model structure affects performance, complementing the benchmark scores.

2.2 Mechanistic Interpretability

Mechanistic interpretability (MI) (Olah et al., 2020; Sharkey et al., 2025) is an emerging subfield of interpretability that aims to understand a neural network model by reverse-engineering its internal computations. Recently, MI has garnered significant attention for interpreting transformer-based LLMs, showing promise in providing insights into the functions of various model components (e.g., neurons, attention heads), offering mechanistic explanations for different model behaviors, and enabling users to optimize the utilization of LLMs (Rai et al., 2024; Luo and Specia, 2024; Zhao et al., 2024).

However, most research on MI has focused on specific components or specialized tasks, without providing a unified explanation of how the overall structure of LLMs relates to their general capabilities. In contrast, our study adopts a data-driven approach: first, by uncovering phenomena through mining structured datasets, and then applying MI techniques to validate these phenomena, we aim to achieve a comprehensive understanding of how model structures and performance interact.

3 LLMs Structure and Performance Dataset

Our dataset is sourced from the Hugging Face model database and the Open LLM Leaderboard. Model structure details are retrieved from structured configuration files of models available on Hugging Face.

For model structural configuration, our dataset primarily includes `size` (model size), `d_model` (hidden dimension), `d_ffn` (FFN intermediate size), `heads` (number of attention heads), `layers` (layer depth), `date` (publication date), and, as an additional feature, `likes` (the number of user likes on Hugging Face model pages).

For model performance, we extract evaluation results from the Open LLM Leaderboard v1, which provides performance metrics for open-source LLMs across six widely used benchmarks : ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), TruthfulQA (Lin et al., 2021), WinoGrande (Sakaguchi et al., 2021), and GSM8K (Cobbe et al., 2021).

The collected data is cleaned and manually verified. Models that are no longer available are removed, and missing data is supplemented through technical reports or source code, ensuring accuracy. Additionally, potential errors are cross-checked during this process. We categorize the models into Mixture of Experts (MoE) and multimodal models. The dataset consists of approximately 160,000 model configuration entries, with roughly 6,000 entries containing performance metrics. The statistical properties of the model structure are summarized in Table 1, while the performance score distribution is shown in Figure 1. The details of the dataset can be found in Appendix A.

Column	Mean	Mode	Q1	Q2	Q3	Max	Skewness	Kurtosis	Miss Rate
size	8	8	1	7	8	1018	12	357	18%
d_model	3284	4096	2048	4096	4096	50257	0	5	5%
d_ffn	12767	14336	9216	14336	14336	13100072	343	120913	21%
heads	28	32	16	32	32	5000	124	32475	5%
layers	30	32	24	32	32	8928	187	49768	5%
kv_heads	15	8	8	8	32	160	1	1	29%
vocab_size	76579	32000	32000	50257	128256	5025700	4	272	4%
pos	30913	4096	2048	4096	32768	104857600	271	85268	7%
downloads	1827	10	10	14	21	24279491	171	36681	5%
likes	2	0	0	0	0	5927	61	5392	5%

Table 1: Statistical summarization of our proposed dataset, includes various statistics for model structure attributes, including **Mean**, **Mode**, **Q1** (first quartile), **Q2** (the middle value of the dataset), **Q3** (third quartile), **Skewness** (measure of asymmetry in the distribution), **Kurtosis** (measure of the "tailedness" of the distribution), and **Miss Rate** (percentage of missing values in the dataset).

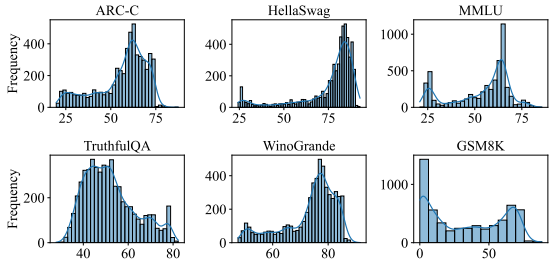


Figure 1: The performance score distributions of open-source LLMs across six benchmarks in our LLMs Structure and Performance Dataset, which illustrate overall performance trends. The x-axis represents performance scores, while the y-axis indicates the number of models achieving each score.

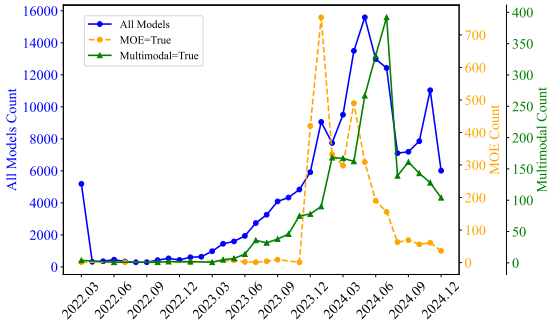


Figure 2: Monthly count distribution of new open-source LLMs: MoE, multimodal, and all models over time.

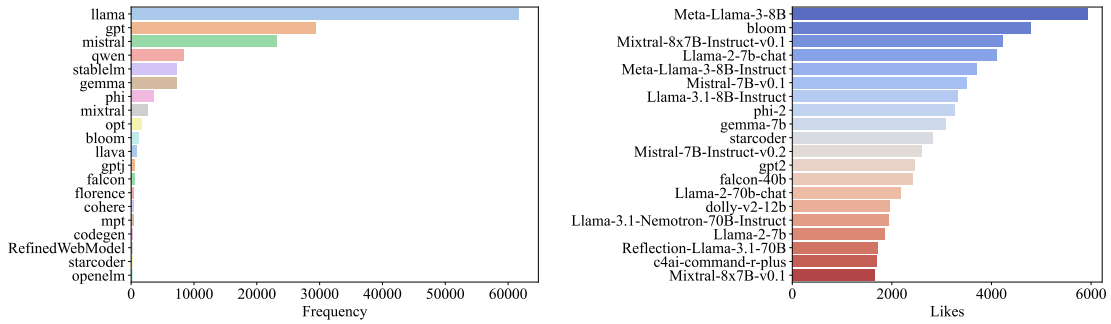
4 Trends Uncovered from Data Analysis

The growth rate of MoE models has slowed, while multimodal models continue to be widely popular. We analyze the monthly variations in the number of LLMs across different categories, as shown in Figure 2. Since the release of ChatGPT in November 2022, the number of LLMs has surged rapidly, followed by a decline in recent months. The trend in multimodal LLMs mirrors that of overall LLMs, as research on multimodal models is often conducted concurrently with base models by the same institutions. In contrast, models based on the MoE architecture saw a sharp increase after the release of Mixtral 8x7B (Jiang et al., 2024) in December 2023. However, its growth rate slowed after six months. Although Deepseek and Qwen have open-sourced smaller models better suited for private deployment (Dai et al., 2024; Yang et al., 2024a), MoE models still require more resources compared to dense models. Additionally, the additional requirements for load balancing result in greater challenges for fine-tuning MoE models, such as instability (Dai et al., 2022).

LLaMA are the most popular base model. Analyzing open-source LLMs model types, such as `NameForCausalLM`, provides insights into the base models used for fine-tuning, as shown in Figure 3a. LLaMA is the most widely adopted base model, followed by the GPT series. Mistral, originating from Europe, ranks third.

7B-scale and 70B-scale models are the most popular. Figure 3b presents the number of likes received by different models. We observe that 7B-scale models are the most popular, offering strong performance while maintaining relatively low resource consumption. Closely following are 70B-scale models, which are highly valued for their exceptional performance.

Models slightly larger than 7B, such as 8B or



(a) (b)

Figure 3: (a) Top 20 types of open-source LLMs sorted by model count. (b) Top 20 open-source LLMs sorted by the number of likes.

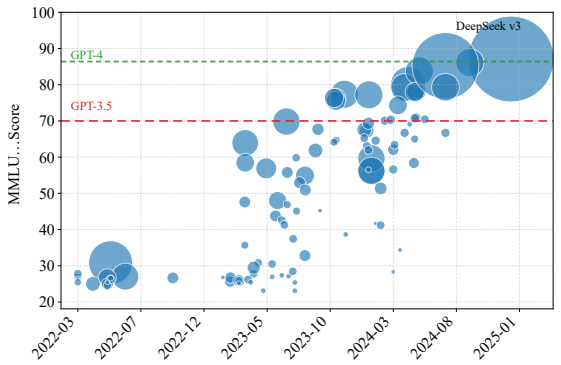


Figure 4: The performance evolution of major open-source pre-trained models in the MMLU over time, where the size of the data points reflects the model scale.

9B, are also well-received. While they exceed the limits of 16GB memory setups, they remain deployable on 24GB platforms like the RTX 3090 or 4090, suggesting a shift in mainstream individual deployment from 16GB to 24GB of VRAM.

The performance of open-source LLMs have steadily improved, and the size of models for achieving the same performance is shrinking. As shown in Figure 4, the release of ChatGPT spurred a surge of new open-source models, accompanied by rapid performance improvements. Over time, these models increasingly rivaled closed-source counterparts. A notable milestone came in December 2024, when Deepseek V3 (Liu et al., 2024) surpassed GPT-4 on the MMLU benchmark.

Meanwhile, the model size required to achieve comparable performance has steadily decreased. In July 2023, matching GPT-3.5 required a 70B model like LLaMA-2-70B (Touvron et al., 2023b), whereas by May 2024, a 9B model such as Yi-1.5-9B (Young et al., 2024) was sufficient.

Different Impact of Model Size and Training Strategy on Task Performance. To analyze the impact of model size and training strategy on performance, we visualize trends in Figure 5. To reduce variance caused by skewed size distribution, we apply equal-frequency binning, ensuring each bin contains the same number of models while adapting to data density. The mean performance score in each bin is used as the representative value, while the interquartile range (IQR) of each bin indicates performance variability.

We observe a general positive correlation between model size and performance for models smaller than 10B and larger than 20B. However, models in the 10B–20B range show an average performance dip. This may be due to differing learning dynamics in this range, though there is no direct evidence. A more plausible explanation is that sub-10B models have been extensively optimized, while 10B–20B models, lacking both scale advantages and popularity, have not reached their full potential.

On the GSM8K benchmark, performance differences across models are more pronounced than on other tasks, highlighting significant disparities in mathematical capability. Improving math performance requires careful model design and optimization. Notably, post-training leads to the largest gains on TruthfulQA, demonstrating its effectiveness in enhancing factual accuracy.

5 Attributing LLMs Performance to Structure Factors

Scores on ARC-C, HellaSwag, and WinoGrande are highly correlated. We compute Spearman rank correlation coefficients (Fieller et al., 1957) to assess performance relationships across datasets (Figure 6). This non-parametric metric ranges from

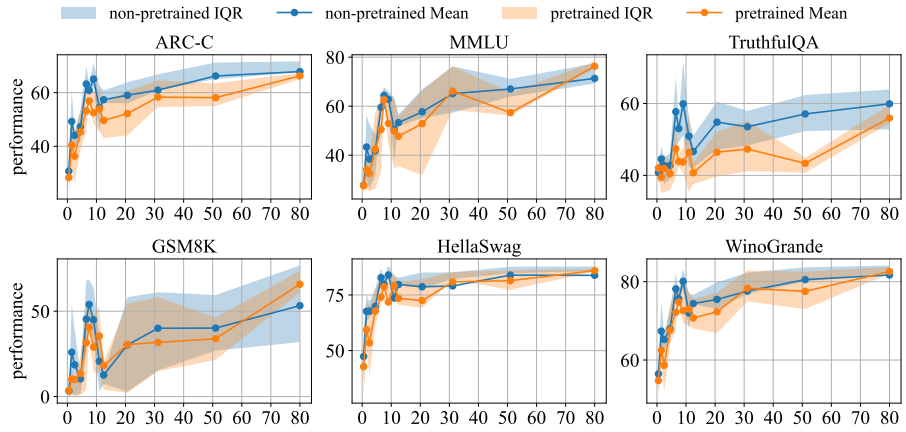


Figure 5: Performance of different datasets across different model size and training strategies, with equal-frequency binning and interquartile range (IQR) shading to capture performance variation.

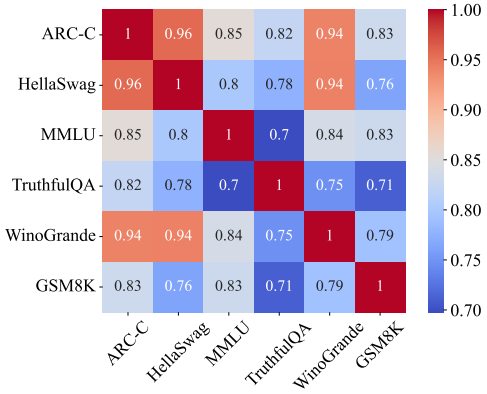


Figure 6: Spearman rank correlation coefficients matrix of performance across different benchmarks.

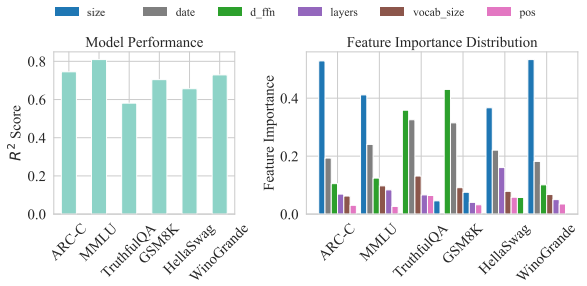


Figure 7: Regression analysis of key parameters and performance across different benchmarks using the Random Forest algorithm, with corresponding R^2 scores and feature importance.

–1 to 1, indicating the strength and direction of monotonic associations. The results reveal strong correlations among ARC-C, HellaSwag, and WinoGrande, likely due to their shared focus on reasoning ability.

Regression analysis demonstrates a significant correlation between model structure, hyperparameters, and performance. We aim to explore the relationship between structure, hyperparameters, and the performance of LLMs. To this end, we selected a set of key parameters and employed various machine learning (ML) algorithms for regression analysis to investigate how these parameters correlate with model performance, including Random Forest (Breiman, 2001), Linear Regression, Decision Tree (Quinlan, 2014), SVR (Cortes, 1995), Ridge (Hoerl and Kennard, 1970), Lasso Regression (Tibshirani, 1996), k-Nearest Neighbors (Kramer and Kramer, 2013), and Gradient Boosting (Friedman, 2001). Especially, we fine-tuned the LLaMA-2-7B model for regression tasks using LLaMA-Factory (Zheng et al., 2024) and LoRA (Hu et al., 2021) techniques, employing a text-based format. The detailed experiment configurations of the models used, along with examples of predictions from the fine-tuned LLaMA-2-7B, can be found in Appendix B.1 and Appendix B.2.

We utilize the R^2 score, also known as the coefficient of determination, to assess the effectiveness of each regression method. R^2 is given by Equation 1:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \quad (1)$$

where y_i are the actual values, $\hat{y}_i$ are the predicted values, and $\bar{y}$ is the mean of the actual values. A higher R^2 indicates a better fit of the model to the data.

The corresponding R^2 scores are shown in Table 2. Machine learning results reveal a clear correlation between model structure and performance,

Model	ARC-C	MMLU	TruthfulQA	GSM8K	HellaSwag	WinoGrande
Random Forest	**75%**	**81%**	**58%**	**70%**	**66%**	**73%**
Linear Regression	52%	54%	32%	44%	41%	50%
Decision Tree	69%	79%	54%	63%	57%	68%
SVR	64%	68%	46%	58%	51%	62%
Ridge	52%	54%	32%	44%	41%	50%
Lasso Regression	52%	54%	32%	44%	41%	50%
k-Nearest Neighbors	71%	77%	50%	67%	62%	69%
Gradient Boosting	72%	78%	56%	67%	64%	71%
MLP	68%	74%	49%	64%	56%	66%
LLM Fine-tune	60%	65%	17%	39%	51%	56%

Table 2: R^2 scores when predicting LLMs' performance across different datasets using key parameters with various methods.

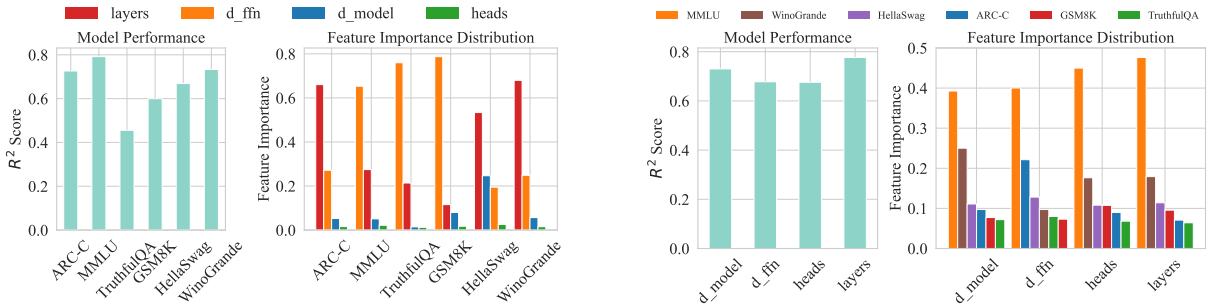


Figure 8: Regression analysis of model structure and performance using Random Forest algorithm. (a) Predicting performance using structure; (b) Predicting structure using performance.

with random forest achieving the highest predictive accuracy. We also compute the Mean Absolute Error (MAE), which remains below 6 for most tasks except GSM8K, indicating practical predictive value. Moreover, the fine-tuned model can reasonably predict performance across benchmarks using a text-based format, suggesting a future where LLMs autonomously analyze data, adapt structures, and evolve to meet new challenges (Tao et al., 2024).

Model size and release date are the primary factors influencing performance. To evaluate the impact of these features, we extracted feature importance from the Random Forest algorithm, which demonstrated the best performance among the tested methods. This feature importance reflects the contribution of each feature in reducing Gini impurity across all tree splits (Genuer et al., 2010). Formally, the feature importance of feature f is given by Equation 2:

$$I_f = \sum_{t \in T} \Delta\text{Gini}(t, f), \tag{2}$$

where T represents the set of all decision trees, and $\Delta\text{Gini}(t, f)$ denotes the decrease in Gini impurity at node t resulting from the use of feature f for splitting.

As presented in Figure 7, we observe that benchmark performance is most strongly correlated with model size and release date. The correlation with model size is relatively straightforward. The release date reflects not only improvements in training techniques but also a steady increase in pretraining token counts: from 1T in LLaMA, to 2T in LLaMA-2, 8T in Mistral (Jiang et al., 2023), and roughly 15T in the latest models (Dubey et al., 2024).

Layer depth and d_{ffn} impact different types of benchmarks. We analyzed key structural variables—`layers` (layer depth), `d_ffn` (FFN intermediate size), `d_model` (hidden dimension), and `heads` (attention heads)—as shown in Figure 8a. Our results suggest that `layers` mainly affects reasoning tasks (e.g., ARC-C, HellaSwag, WinoGrande), while `d_ffn` more strongly influences mathematical ability and knowledge accuracy, as

seen in GSM8K, MMLU, and TruthfulQA. Experiments analyzing the impact of developer proficiency and development timing (Appendix C.1) reinforce the robustness and generalizability of our findings.

This aligns with prior analyses: layer depth governs the degree of non-linearity, thereby enhancing reasoning abilities (Jin et al., 2024; Mueller and Linzen, 2023; Ye et al., 2024), whereas empirical studies indicate that LLMs store knowledge mainly in the FFNs (Geva et al., 2020; Stolfo et al., 2023), with larger d_{ffn} substantially boosting memory capacity. This also concurs with findings that increasing the number of experts in MoE models—viewed as an extension of the FFNs—improves performance on knowledge-intensive tasks but not on reasoning (Jelassi et al., 2024; Fedus et al., 2022).

Furthermore, Mirzadeh et al. (2024) observe that even minor modifications to the GSM8K dataset cause a significant performance drop, suggesting that current LLMs primarily rely on memorization to solve mathematical problems. Meanwhile, Stolfo et al. (2023) find that LLMs mainly execute basic arithmetic operations within the FFNs. Together, these studies explain why d_{ffn} plays a more critical role than layer depth on the GSM8K task.

Additionally, we collected extra performance data on BigCodeBench and IFEval to address the incomplete task coverage. We then conducted regression analysis on these datasets using random forest models, achieving R^2 scores of 66.2% and 48.2%, respectively. Feature importance scores, presented in Table 3, indicate that layer depth is the most influential factor for coding-related tasks, likely due to their reasoning-intensive nature. In contrast, instruction-following tasks are more sensitive to the capacity of FFNs, with d_{ffn} identified as the dominant contributor.

Benchmark	layers	d_model	d_ffn	heads
BigCodeBench	35.4%	33.5%	23.1%	8.1%
IFEval	29.3%	25.2%	39.0%	6.5%

Table 3: Feature importance of structural configurations in random forest regression models for BigCodeBench and IFEval.

MMLU is the most representative benchmark. Our analysis reveals that MMLU performance is the key feature for predicting model structure, as shown by the feature importance values in Figure 8b. This supports the hypothesis that MMLU scores best capture overall model performance and aligns with how organizations like OpenAI, Anthropic, Mistral, and Qwen typically showcase model capabilities on MMLU.

6 Mechanistic Interpretability Analysis

6.1 Validating the Impact of Layer Depth via Layer Pruning

We apply the ShortGPT (Men et al., 2024) method to prune LLaMA-2-7B to validate the impact of layer depth. The experiments on the Qwen-2-7B models are shown in Appendix C.3. By pruning a small number of less important layers, we aim to minimize disruption to the model's overall capabilities while allowing us to observe how changes in depth affect performance across various tasks. To identify these layers, we use the Block Influence (BI) metric, defined for the i^{th} layer is given by Equation 3:

$$\mathrm{BI}_i = 1 - E_{X,t}\frac{X_{i,t}^T X_{i+1,t}}{\|X_{i,t}\|_2 \|X_{i+1,t}\|_2}, \qquad (3)$$

where $X_{i,t}$ is the t^{th} row of the hidden state at layer i. A lower BI score indicates higher cosine similarity between X_i and X_{i+1}, suggesting that the layer contributes less transformation and is thus less critical.

By averaging BI scores over multiple benchmarks for the LLaMA-2-7B model, we observe consistent patterns across layers, as shown in Appendix C.2, making it challenging to use BI scores alone to differentiate the functional roles of individual layers across tasks. Therefore, we prune layers 21 through 29, which have the lowest BI scores. We also find that BI scores tend to be higher in early and final layers, and lower in middle-to-later layers, consistent with prior work (Kim et al., 2024).

We observe an anomaly in the GSM8K benchmark, which requires models to generate precise numerical answers rather than selecting from multiple choices as in other benchmarks. This unique task structure makes GSM8K not directly comparable to the others. Therefore, we exclude GSM8K from this experiment.

After pruning these layers, we evaluate the model using lm-evaluation-harness (Gao et al., 2024) following *the leaderboard* protocols, comparing its performance before and after pruning across multiple benchmarks. The results are shown in Figure 9.

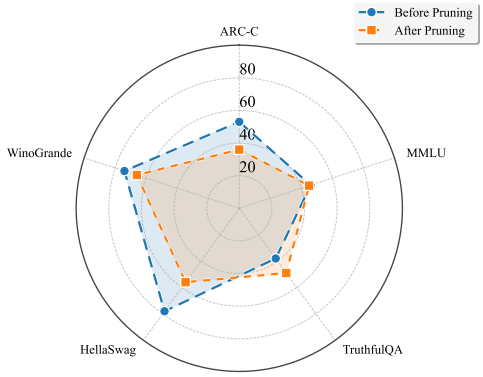


Figure 9: Performance across different benchmarks of Llama-2-7B before and after pruning 21-29 layers.

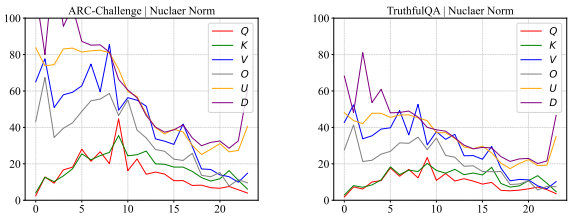


Figure 10: Layer-wise gradient analysis during fine-tuning of Qwen-2-0.5B on the ARC-C and TruthfulQA benchmarks.

Pruning leads to significant performance drops on benchmarks where layer depth is a critical factor (ARC-C, HellaSwag, WinoGrande), confirming the random forest regression results (Figure 8a). Conversely, benchmarks less dependent on layer depth (e.g., MMLU, TruthfulQA) show minimal degradation, with TruthfulQA even improving slightly, further validating our analysis.

6.2 Validating Findings through Layer-wise Gradient Analysis

Following the gradient analysis methodology of Li et al. (2024), we evaluate the gradients during fine-tuning of Qwen-2-0.5B on the ARC-C and TruthfulQA benchmarks, which are representative tasks where layers depth and d_{ffn}, respectively, are identified as the most influential structural factors.

Our analysis focuses on six major weight matrices in each decoder layer: the Query (Q), Key (K), Value (V), and Output (O) projections in the attention module, as well as the Up (U) and Down (D) projections in the FFN module. We denote $X \in \{Q, K, V, O, U, D\}$.

The loss L_θ corresponds to the cross-entropy loss for next-token prediction used in supervised fine-tuning, where only the response tokens contribute to the overall loss, and instructions are ignored. We perform multiple backward passes until gradients from all entries in the dataset are accumulated.

For the weight matrix X_i of the i-th layer and its corresponding gradient $G_{X,i}$, we measure the concentration of its gradient spectrum on dominant singular values using the Nuclear Norm $s_{X,i}$. This provides insights into the gradient behavior across different layers and tasks. The Nuclear Norm is given by Equation 4:

$$s_{X,i} = \|G_{X,i}\|_* = \sum_{j=1}^{\min(m,n)} |\sigma_j|, \tag{4}$$

where σ_j denotes the j-th singular value, computed via singular value decomposition (SVD), as shown in Equation 5:

$$\begin{aligned} \Sigma &= \operatorname{diag}\left(\sigma_1, \sigma_2, \cdots, \sigma_{\min(m,n)}\right), \\ G_{X,i} &= U \Sigma V^\top. \end{aligned} \tag{5}$$

The results of this analysis are shown in Figure 10. We observe that gradients in the deeper layers of the ARC-C benchmark remain relatively high, indicating that deeper layers play a more critical role in successfully completing reasoning tasks. This finding aligns with our earlier observation that layer depth is the key structural factor for ARC-C. In contrast, gradients in the deeper layers of the TruthfulQA benchmark are substantially lower, suggesting that these layers contribute less to this memory-centric task.

The experiment on LLaMA-3.2-3B is presented in Appendix C.4. Meanwhile, a deeper investigation into the gradient dynamics, as detailed in Appendix C.5, further supports this hypothesis.

7 Conclusion

This study provides a comprehensive, data-driven analysis of LLMs through a large-scale dataset that captures structural configurations and their performance across diverse benchmarks. By systematically tracing the evolution of LLMs, we identify emerging trends and offer insights into future directions. Our findings underscore the critical influence of structural configurations on model performance, validated through mechanistic interpretability techniques. This work delivers actionable, data-driven guidance for optimizing LLM design, paving the way for the development of more efficient, scalable, and adaptable models to meet the demands of diverse real-world applications.

Limitations

This study focused on a specific set of tasks, potentially limiting the generalizability of our findings. Different applications may involve distinct requirements and data characteristic. Future work should explore a broader range of tasks to improve the robustness and applicability of our conclusions.

Our mechanistic interpretability analysis was limited to methods such as layer pruning and gradient analysis. While these techniques provided valuable insights, they may not fully capture the complex internal dynamics of LLMs. Future research could incorporate a wider variety of interpretability tools to validate and complement our findings, thereby offering a more comprehensive understanding of model behavior.

Ethics Statement

All training and evaluation datasets used in this study are publicly available under open-access licenses and intended solely for research purposes. These datasets contain no personal or identifiable information, nor any offensive content. The data analyzed in this work pertains exclusively to model structure and performance metrics.

All datasets developed or used in this research will be released under the MIT License. We share these resources to promote transparency, reproducibility, and further research within the community. We encourage others to build upon and improve our work, provided they adhere to the terms of the MIT License.

References

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

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

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Corinna Cortes. 1995. Support-vector networks. *Machine Learning*.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Stablemoe: Stable routing strategy for mixture of experts. *Preprint*, arXiv:2204.08396.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Edgar C Fieller, Herman O Hartley, and Egon S Pearson. 1957. Tests for rank correlation coefficients. i. *Biometrika*, 44(3/4):470–481.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. 2010. Variable selection using random forests. *Pattern recognition letters*, 31(14):2225–2236.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Samy Jelassi, Clara Mohri, David Brandfonbrener, Alex Gu, Nikhil Vyas, Nikhil Anand, David Alvarez-Melis, Yuanzhi Li, Sham M Kakade, and Eran Malach. 2024. Mixture of parrots: Experts improve memorization more than reasoning. *arXiv preprint arXiv:2410.19034*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. 2024. Exploring concept depth: How large language models acquire knowledge at different layers? *arXiv preprint arXiv:2404.07066*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. 2024. Shortened llama: A simple depth pruning for large language models. *arXiv preprint arXiv:2402.02834*, 11.

Oliver Kramer and Oliver Kramer. 2013. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*, pages 13–23.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Ming Li, Yanhong Li, and Tianyi Zhou. 2024. What happened in llms layers when trained for fast vs. slow thinking: A gradient perspective. *CoRR*, abs/2410.23743.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Haoyan Luo and Lucia Specia. 2024. From understanding to utilization: A survey on explainability for large language models. *arXiv preprint arXiv:2401.12874*.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Aaron Mueller and Tal Linzen. 2023. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. *arXiv preprint arXiv:2305.19905*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

J Ross Quinlan. 2014. *C4. 5: programs for machine learning*. Elsevier.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. 2025. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. *arXiv preprint arXiv:2305.15054*.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. 2024. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Haoran Yang, Yumeng Zhang, Jiaqi Xu, Hongyuan Lu, Pheng Ann Heng, and Wai Lam. 2024b. Unveiling the generalization power of fine-tuned large language models. *arXiv preprint arXiv:2403.09162*.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2024. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. *arXiv preprint arXiv:2407.20311*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

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

H Zhao, F Yang, B Shen, and HLM Du. 2024. Towards uncovering how large language model works: An explainability perspective. *arXiv preprint arXiv:2402.10688*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Appendices

A Details of the LLMs Structure and Performance Dataset

A.1 Detailed Description of Each Column

As shown in Table 4, each column presents key metrics and attributes of the model, offering valuable insights into characteristics such as its size, structure, and usage statistics.

Column	Name	Unit	Description
size	Model Size	Billions	The overall parameter count of the model.
d_model	Hidden Dim	1	The size of the hidden state of the model. Usually describing how wide the model is.
d_ffn	Intermediate Size	1	The size of the intermediate state of the MLP (or GLU) in the FFN of each Transformer Decoder Layer. A wider model usually has a larger d_ffn.
heads	Attention Head Count	1	The number of attention heads.
layers	Decoder Layer Count	1	The number of Decoder layers. A deeper model is whose layer count is larger.
kv_heads	KV Head Count	1	The number of KV heads. Related with GQA (MQA) and the size of KV cache per token. Equal to the heads count for MHA, 4 to 16 times smaller for GQA variant.
vocab_size	Vocabulary Size	1	The available token count of the tokenizer, as well as the embedding and LM_head component of the base model. Larger vocab means less sequence length, more efficient in inference but at the cost of more parameter.
pos	Maximum Input Position	1	The maximum capable input sequence length. Relate with sin and cos value caching of Rotary Positional Embedding, also indicating the long context ability with the model.
downloads	Download Count	1	The download count on Hugging Face model pages, reflecting actual usage and interest from the community.
likes	Like Count	1	Users' like count on Hugging Face model pages, reflecting community recognition.

Table 4: Description of each column from our LLMs Structure and Performance Dataset.

A.2 The example of the LLMs Structure and Performance Dataset

As shown in Table 5, the structure parameters of several models and their performance across different benchmarks are presented, including Llama-3-8B, Bloom (Le Scao et al., 2023), Mixtral-8x7B, Llama-2-7B, and Mistral-7B.

B Experimental Details

B.1 Resources Used in the Experiments

All experiments were conducted on two RTX 4090 GPUs, utilizing a total of 200 GPU hours. The tasks included regression analysis of model structure and performance, fine-tuning the LLaMA-2-7B model for regression tasks using the Low-Rank Adaptation (LoRA) technique and the Llama-Factory framework, pruning specific layers of the LLaMA-2-7B model, and evaluating the model on ARC-C, TruthfulQA, WinoGrande, HellaSwag, and MMLU benchmarks using the lm-evaluation-harness.

https://github.com/hiyouga/LLaMA-Factory
https://github.com/EleutherAI/lm-evaluation-harness

Parameter	Llama-3-8B	bloom	Mixtral-8x7B	Llama-2-7B	Mistral-7B
size	8	176	46	7	7
d_model	4096	14336	4096	4096	4096
d_ffn	14336		14336	11008	14336
heads	32	112	32	32	32
layers	32	70	32	32	32
kv_heads	8		8	32	8
vocab_size	128256	250880	32000	32000	32000
pos	8192		32768	4096	32768
likes	4883	4632	3920	3633	3259
downloads	556210	28821	2911366	927400	3147345
ARC-C	60.24	50.43	66.38	53.07	59.98
HellaSwag	82.23	76.41	86.46	78.59	83.31
MMLU	66.7	30.85	71.88	46.87	64.16
TruthfulQA	42.93	39.76	46.81	38.76	42.15
WinoGrande	78.45	72.06	81.69	74.03	78.37
GSM8K	45.19	6.9	57.62	14.48	37.83

Table 5: Examples from our LLMs Structure and Performance Dataset.

Additionally, we performed gradient analysis during the fine-tuning of the Qwen-2-0.5B model on the ARC-C and TruthfulQA benchmarks.

B.2 Hyperparameter Configuration for Regression Models

For regression analysis of model structure and performance, various models were employed. The hyperparameter configurations for these models are provided in Table 6.

The LLaMA-2-7B model was fine-tuned using a text-based format, where the model takes a different structure as input and predicts performance across multiple datasets. As shown in Figure 11, the fine-tuned model demonstrates strong performance in accurately predicting outcomes in the specified text format.

Model	Hyperparameters
Random Forest	random_state=42, n_estimators=100, max_depth=None
Linear Regression	fit_intercept=True, normalize=False
Decision Tree	random_state=42, max_depth=None, min_samples_split=2
SVR	kernel=rbf, C=1.0, epsilon=0.1
Ridge	alpha=1.0, fit_intercept=True
Lasso Regression	alpha=0.1, max_iter=1000
k-Nearest Neighbors	n_neighbors=5, algorithm=auto
Gradient Boosting	n_estimators=100, learning_rate=0.1, max_depth=3
XGBoost	objective=reg:squarederror, n_estimators=100, learning_rate=0.1
MLP	hidden_layer_sizes=(32, 64, 32), max_iter=100, activation=relu
LLM Fine-tune	lora_target=all, learning_rate=1.0e-4, num_train_steps=3500

Table 6: Regression models and their key hyperparameters.

Examples of Performance Regression Prediction using Fine - tuned Llama2 7B Model

Prompt1: You are an AI model expert. Analyze the model structure and predict performance metrics. Model Architecture: Num attention heads: 32, Num hidden layers: 32, Vocab size: 32000, Max position embeddings: 32768, Year: 2024, Month: 1, Day: 3, Model dimension: 4096, FFN hidden dimension: 14336, Model parameters: 7.000B
Truth1:
Prediction: ARC-C: 55.20, HellaSwag: 78.22, MMLU: 50.30, TruthfulQA: 57.08, WinoGrande: 73.24, GSM8K: 11.45
Answer1:
Prediction: ARC-C: 67.41, HellaSwag: 86.78, MMLU: 64.07, TruthfulQA: 67.68, WinoGrande: 81.61, GSM8K: 59.74

Prompt2: You are an AI model expert. Analyze the model architecture and predict performance metrics. Model Architecture: Num attention heads: 40, Num hidden layers: 36, Vocab size: 50688, Max position embeddings: 2048, Year: 2023, Month: 2, Day: 27, Model dimension: 5120, FFN hidden dimension: 20480, Model parameters: 12.000B
Truth2:
Prediction: ARC-C: 41.38, HellaSwag: 70.26, MMLU: 25.63, TruthfulQA: 33.00, WinoGrande: 66.46, GSM8K: 1.44
Answer2:
Prediction: ARC-C: 46.42, HellaSwag: 70.00, MMLU: 26.19, TruthfulQA: 39.19, WinoGrande: 62.19, GSM8K: 0.61

Figure 11: Performance prediction examples using a fine-tuned Llama-2-7B model.

C Further Experiment Result

C.1 Analyzing the Impact of Developer Proficiency and Development Timing

The central goal of our study is to uncover unified relationships between model structure and performance through large-scale data mining over structural datasets. Due to the breadth and diversity of our dataset, we expect that secondary factors exert minimal influence on the extracted conclusions, as core patterns can be robustly identified across a wide range of models.

Nevertheless, to ensure that our experimental conclusions are not affected by differences in the development proficiency of various model providers, and to mitigate the possibility that our analysis is overly skewed toward LLaMA-based models, we aimed to achieve broader model representation beyond LLaMA-based architectures while maintaining high model quality.

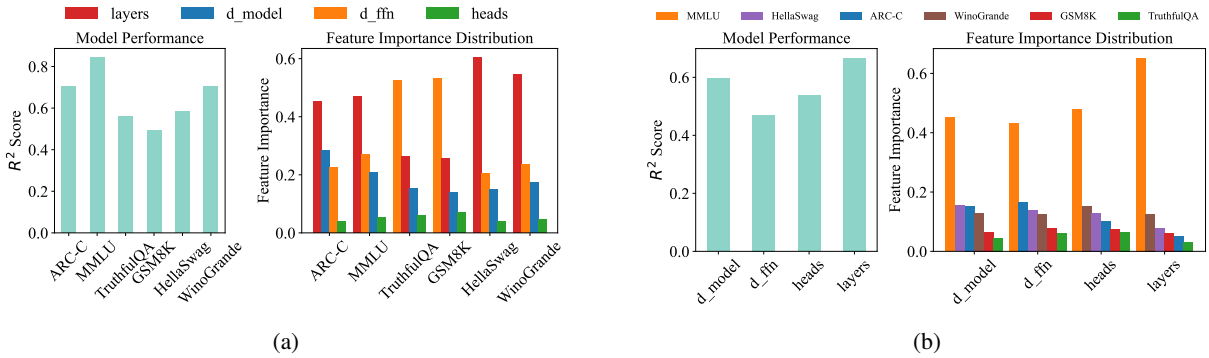
To this end, we selected models from Hugging Face's `open-llm-leaderboard/official-providers` (e.g., LLaMA, MistralAI, DeepSeek, Qwen), which are known to follow high-quality training standards. This filtering process resulted in a dataset where LLaMA-based models and their variants comprised only 27% of the total, effectively reducing potential bias due to their overrepresentation.

As shown in Figure 12a, our results remained consistent with earlier findings: layer depth emerged as the most important structural parameter for ARC-C, HellaSwag, and WinoGrande, while d_{ffn} was most critical for TruthfulQA and GSM8K. MMLU was the only exception, likely due to data sparsity.

Meanwhile, as shown in Figure 12b, performance on the MMLU dataset was identified as the most important parameter for predicting the model's architectural configuration, which aligns with previous conclusions.

To avoid the impact of temporal variations, we augmented our Random Forest regression model with the date variable. As shown in Figure 13, the resulting R^2 scores and feature importance indicate that structural features continue to be significant even when accounting for temporal effects, supporting our

conclusion that benchmarks like ARC-C, HellaSwag, and Winogrande rely heavily on model depth. In contrast, d_{ffn} emerges as the dominant factor for MMLU, GSM8K, and TruthfulQA.



(a) (b)

Figure 12: Regression analysis of major high-quality model structure parameters and their performance across benchmarks using the Random Forest algorithm. (a) Predicting performance from model structure; (b) Predicting model structure from performance.

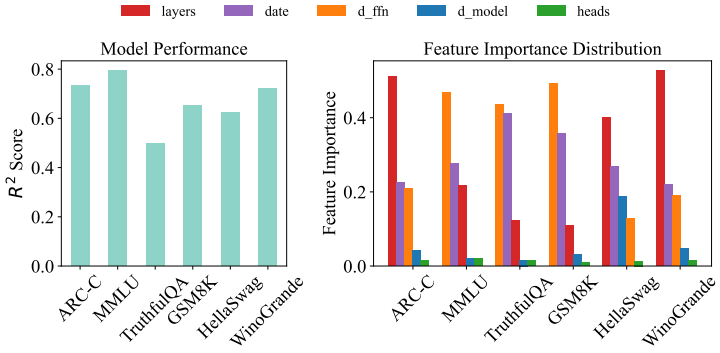


Figure 13: Feature importance in the Random Forest model with date included. Structural features like depth and d_ffn remain dominant despite temporal effects.

C.2 Analysis of BI Scores Across Layers in the LLaMA-2 7B Model across Different Benchmarks

As shown in Figure 14, we present the BI scores for different layers of the LLaMA-2-7B model across various benchmarks. The analysis highlights the relative contribution of each layer to model performance on tasks from diverse domains.

C.3 Layer Pruning Analysis with Qwen-2-7B

Similar to the pruning experiments conducted on LLaMA-2-7B, we also prune the Qwen-2-7B model and observe consistent conclusions, as illustrated in Figure 15. Pruning leads to significant performance drops on benchmarks where layer depth is a critical factor (e.g., ARC-C, HellaSwag, WinoGrande), confirming the findings of the random forest regression analysis. Conversely, benchmarks that are less dependent on layer depth (e.g., MMLU, TruthfulQA) exhibit minimal performance degradation, further validating our analysis.

C.4 Layer-wise Gradient Analysis with LLaMA-3.2-3B

Similar to the layer-wise gradient analysis conducted on Qwen-2-0.5B, we performed the same experiment on LLaMA-3.2-3B, as shown in Figure 16, and found results consistent with our original conclusions. We observe that gradients in the deeper layers of the ARC-C benchmark remain relatively high, while gradients in the deeper layers of the TruthfulQA benchmark are substantially lower. These results further support our previous conclusions.

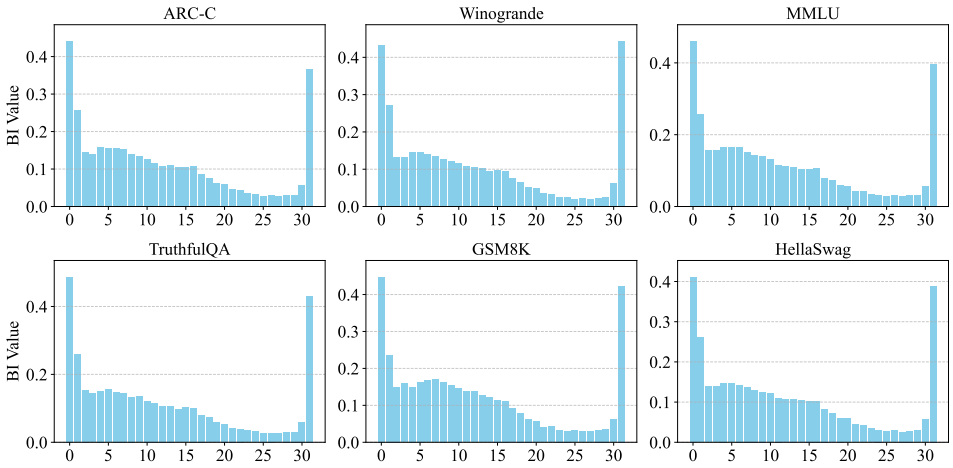


Figure 14: BI scores of different layers in the LLaMA-2-7B model across various benchmarks.

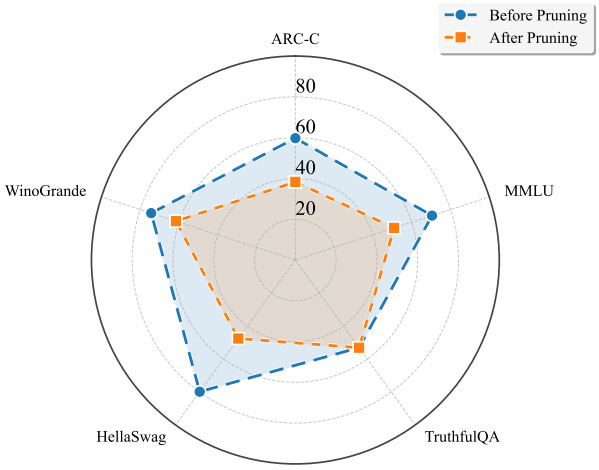


Figure 15: Performance across different benchmarks of Qwen-2-7B before and after pruning 21-25 layers.

C.5 Layer-wise Gradient Analysis with Different Language Styles

We further explore the dynamics of different layers within the model, particularly the deeper layers, to explain how task dependencies vary with model depth. Following the methodology in Section 6.2, we conducted gradient analysis across different corpora. Our findings, shown in Figure 17, reveal a significant increase in gradients within the deeper FFN layers when the model encounters distinct linguistic styles or archaic texts. In contrast, for corpora such as plain text or mathematical data, these layers do not exhibit such anomalous gradient behavior.

We observed that the layers responsible for generating the additional gradient peaks largely correspond to the layers excluded in the previous section. Larger gradients typically suggest insufficient training of the corresponding model components. This implies that layers with large gradients in LLMs process language-form-related components, rather than knowledge components abstracted from linguistic forms. In other words, the increased gradient magnitude reflects a lower retention of knowledge within these layers, explaining the insensitivity of knowledge-based tasks to layer removal. Conversely, reasoning processes are closely tied to language itself, meaning the removal of these layers has a more significant impact on such tasks.

D Explanation of Industry-Specific Jargons

We provide detailed explanations for potentially confusing industry-specific jargon mentioned in the paper, ensuring clarity without compromising technical accuracy.

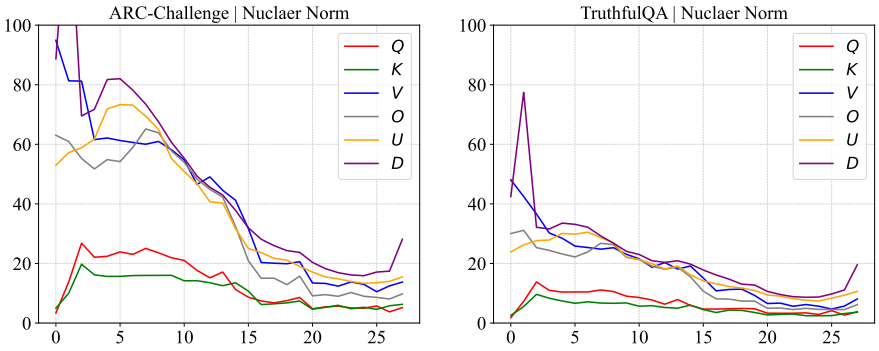


Figure 16: Layer-wise gradient analysis during fine-tuning of LLaMA-3.2-3B on the ARC-C and TruthfulQA benchmarks.

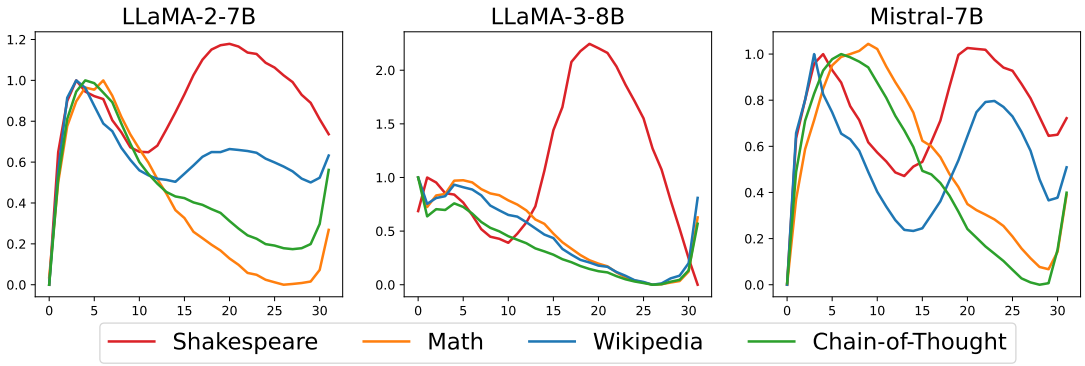


Figure 17: Layer-wise gradient on different corpuses.

The Leaderboard: A standardized platform (e.g., Hugging Face's Open LLM Leaderboard) for comparing model performance across benchmarks.

MoE (Mixture of Experts): A neural network architecture that dynamically routes inputs to a subset of specialized expert models, improving computational efficiency and scalability in large language models (LLMs).

VRAM (Video Random Access Memory): The GPU's dedicated memory, critical for deploying large language models (LLMs) because its capacity constrains the maximum size of models that can be loaded and run.

IQR (Interquartile Range): A statistical measure of data spread between the 25th and 75th percentiles, reducing the influence of outliers. Applied in Figure 5 to capture performance fluctuations across model sizes.

LLaMA-Factory: An open-source framework designed for fine-tuning, training, and deploying large language models.

LoRA (Low-Rank Adaptation): A parameter-efficient fine-tuning technique that uses low-rank matrix decomposition.

Gini Impurity: A measure of impurity in a dataset used in decision tree algorithms to determine the best feature splits by evaluating class distribution at a node.