From Parameters to Performance: Diving into LLM Development and Structure

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved remarkable success across various domains, driving significant technological advancements and innovations in applications. Despite the rapid growth in model scale and capability, systematic research on how structural configurations affect performance remains limited. To address this gap, we present a large-scale dataset encompassing different pen-source LLM structures and their performance across multiple benchmarks. Furthermore, we provide a systematic analysis of the dataset from a data mining perspective. We begin by reviewing the historical development of LLMs and discussing potential future trends. We then investigate the impact of various structural configurations on performance across different benchmarks. Finally, we employ mechanistic interpretability techniques to validate our findings mined from the dataset. Our goal is to provide data-driven insights for optimizing LLMs and offer valuable guidance for 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 LLM 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 LLM 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 LLM 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 LLM Structure and Performance Dataset:** We introduce a large-scale dataset containing a variety of open-source LLM 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 LLM 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 model depth on performance across different benchmarks, as mined from the LLM 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 (Cobbe et al., 2021; Hendrycks et al., 2020; Zellers et al., 2019; Sakaguchi et al., 2021; Lin et al., 2021; Clark et al., 2018). *The leaderboard* hosts a wide array of NLP tasks. By referencing *the leaderboard*, researchers can assess the effectiveness of their models and stay up-to-date with the latest developments in the field.

Although *the leaderboard* provides practical performance comparisons between LLMs, it overlooks the structural information of the models. While the datasets used for benchmarking are diverse, there has been limited exploration of the relationships between these datasets and the different model structure configurations. Our work aims to address this gap by combining model meta-information with performance data from *the leaderboard*. This additional dimension provides valuable insights into how model design has evolved in recent years, 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 mechanistic interpretability 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 mechanistic interpretability techniques to validate these phenomena, we aim to achieve a comprehensive understanding of how model structures and performance interact.

3 LLM Structure and Performance Dataset

Our dataset is sourced from the Hugging Face model database and the Open LLM Leaderboard (Beeching et al., 2023). Model structure details are retrieved from structured configuration files of models available on Hugging Face, typically found in the `config.json` file.

Each entry in our dataset includes the model's size, activation functions, hidden dimension, feed-forward network (FFN) intermediate size, attention head count, layer count, vocabulary size, context length, publication date, and, as an additional feature, the likes count.

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

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).

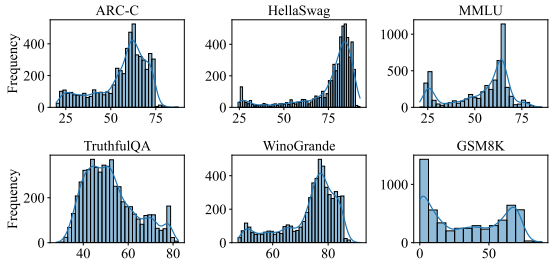


Figure 1: Performance distribution of open-source LLMs across different benchmarks.

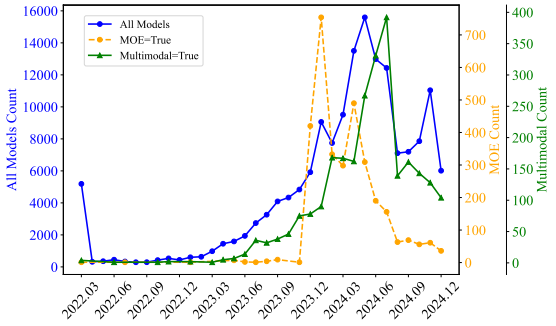


Figure 2: Monthly count distribution of new open-source LLMs: MoE, multimodal, and all models over time.

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.

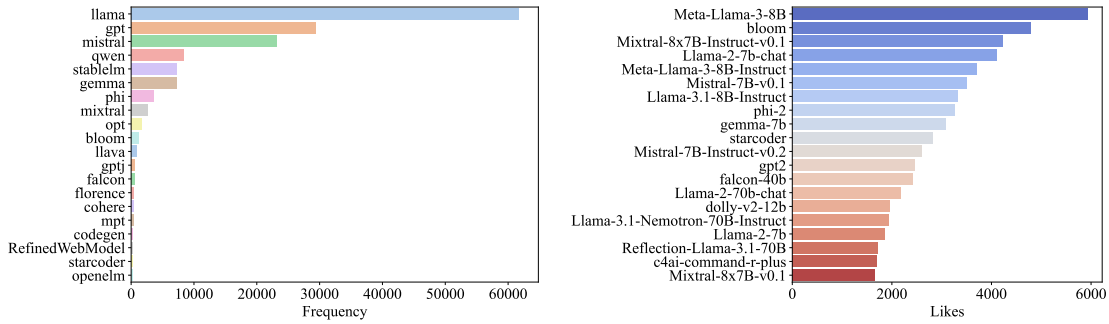
4 Trends Uncovered from Data Analysis

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 newly released models 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. 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 and higher barriers for fine-tuning MoE models, such as instability, and demand higher optimization of training code (Dai et al., 2022; Fedus et al., 2022).

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.

LLaMAs are the most popular base models. Although modern open-source LLMs are largely interoperable at the code level, analyzing their 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 and 70B models are the most popular. Fig-



(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.

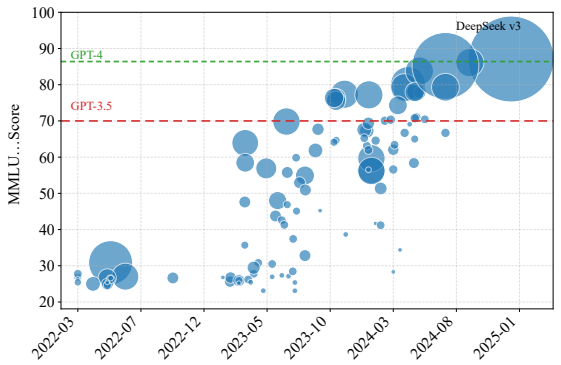


Figure 4: The performance evolution of major pen-source pre-trained models in the MMLU over time, where the size of the data points reflects the model scale.

ure 3b presents the number of likes received by different models. We observe that the 7B model is the most popular, achieving strong performance while maintaining relatively low resource consumption. Following closely are models with 70B parameters, which are highly valued for their exceptional performance. It is also noteworthy that slightly larger models in the 7B range, such as 8B or 9B, are similarly well-received. Although these models cannot be run on traditional 16GB memory setups (unlike the 7B model), they can be deployed on 24GB memory platforms, such as the RTX-3090 or 4090. This may suggest that for individual enthusiasts, mainstream deployment platforms have shifted from 16GB to 24GB of VRAM.

The performance of open-source models has steadily improved, and the size of large models for achieving the same performance is shrinking. As shown in Figure 4, following the release of ChatGPT, a large number of new open-source models were launched. Model performance saw rapid advancements during this period of growth. As the development of open-source models progressed, their performance increasingly rivaled that of closed-source models, particularly those from the GPT series. Notably, in December 2024, the release of Deepseek V3 (Liu et al., 2024) marked a significant milestone, surpassing GPT-4's performance on the MMLU benchmark.

At the same time, the model size required to achieve equivalent performance continues to decrease. For example, to match the performance of GPT-3.5, a 70B parameter model, such as Llama-2-70B (Touvron et al., 2023b), was needed in July 2023, whereas by May 2024, a 9B parameter model, like Yi-1.5-9B (Young et al., 2024), sufficed.

Anomalous impact of model size on task performance. To assess the effect of pretraining on model performance across different datasets and model sizes, we analyze the data presented in Figure 5. We employ equal-frequency binning of model parameters to reduce statistical variance, using the bin's average as the data point. Additionally, we visualize the interquartile range (IQR) to capture performance fluctuations.

We observed that within the domains of models smaller than 10B and larger than 20B, model performance is generally positively correlated with size. However, within the range of 10B to 20B, there is an average decline in performance. One possible explanation is that the learning patterns of models in this size range differ from those of both smaller and larger models, resulting in a lower overall model capability. However, there is no supporting evidence for this hypothesis. A more likely explanation is that models under 10B have been highly optimized and trained (possibly even overfitted), whereas models in the 10B to 20B range are less popular and, not having a significant parameter

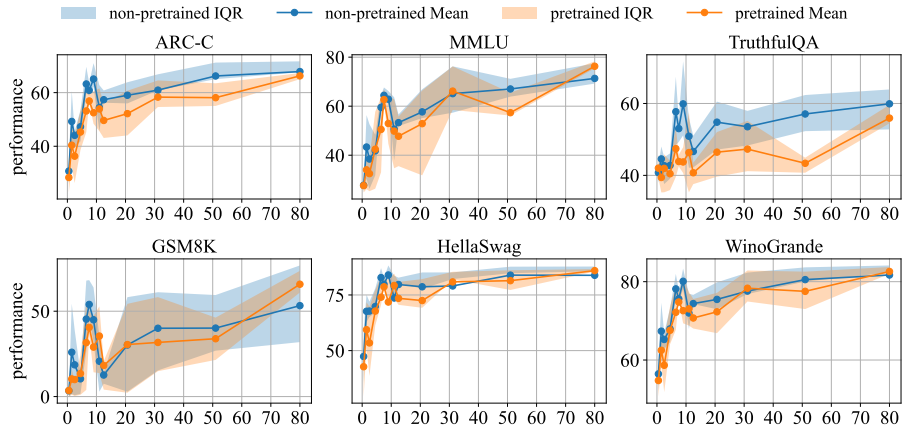


Figure 5: Performance of different datasets across different model size and training strategies, with equal-frequency binning and interquartile range (IQR) shading to capture performance variation.

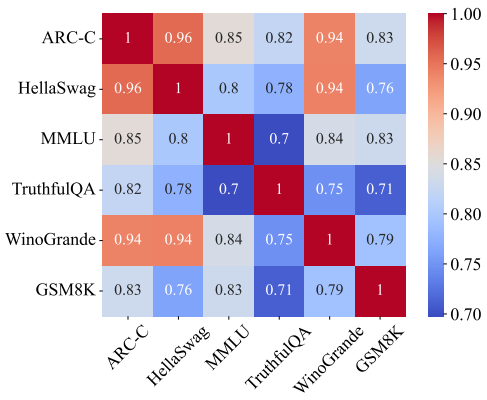


Figure 6: Spearman rank correlation coefficients matrix of performance across different benchmarks.

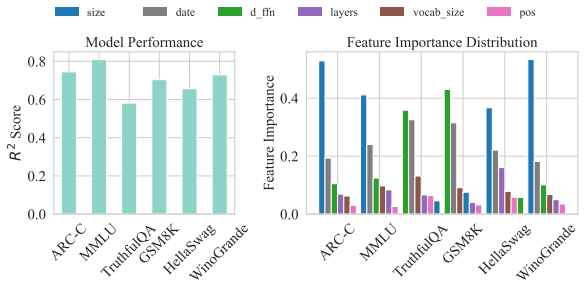


Figure 7: Regression analysis of key parameters and performance across different benchmarks using the Random Forest algorithm, with corresponding R^2 scores and feature importance.

advantage, have not reached the performance limits of their larger counterparts.

Additionally, we noted that for the GSM8K benchmark, the performance variation between different models is more pronounced compared to other benchmarks. The disparity in mathematical capabilities across different models is significant. To enhance mathematical performance, careful model design and optimization are essential. Additionally, post-training yields the greatest performance improvement on TruthfulQA, demonstrating its effectiveness in enhancing the accuracy of large model knowledge.

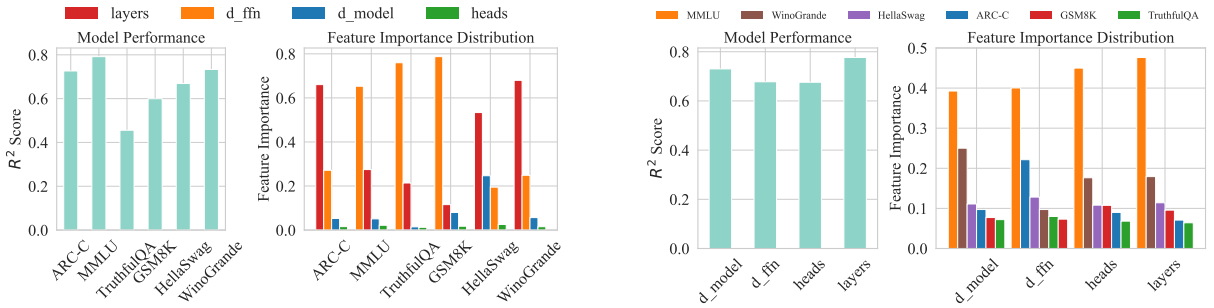
5 Attributing LLM Performance to Structure Factors

The scores on the ARC-C, HellaSwag, and WinoGrande datasets are highly correlated. We calculated the Spearman rank correlation coefficients (Fieller et al., 1957) to assess the relationship between model performance across these datasets, as shown in Figure 6. The Spearman rank correlation coefficient is a non-parametric measure of the strength and direction of association between two ranked variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). The results indicate a particularly high correlation in performance between the ARC-C, HellaSwag, and WinoGrande datasets. These datasets primarily assess the model's reasoning abilities, which may explain the strong correlation in the performance scores.

Regression methods indicate that model structure can be used to predict model performance. We aim to explore whether the structure characteristics of a model can help predict its performance. To assess the relative importance of model size, parameters, and release date on performance across different benchmarks, we employed various machine learning (ML) algorithms for regression, including

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.



(a) (b)

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

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). Specifically, 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.2 and Appendix B.1.

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

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

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 presented in Table 2. The results of the machine learning approaches demonstrate a clear correlation between model structure and performance, with the random forest method achieving the highest predictive accuracy. Additionally, the fine-tuned model can leverage its structure to predict performance across diverse benchmarks with reasonable accuracy using a text-based format. This suggests a future direction where large models autonomously analyze data, refine their structure, and adapt to new challenges, enabling self-evolution (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). Specifically, it is calculated by assessing the reduction in Gini impurity caused by each feature at every split in the decision trees. Formally, the

feature importance of feature f is given by:

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

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.

From the data presented in Figure 7, we observe that benchmark performance is most strongly correlated with model size and release date. Among these, the correlation with model size is trivial. The release date, particularly for modern LLMs, reflects not only advancements in training techniques but also a progressive increase in the number of tokens used during the pre-training phase. For instance, this has grown from 1T tokens in LLaMA (Touvron et al., 2023a) to 2T in LLaMA-2 (Touvron et al., 2023b), 8T in Mistral (Jiang et al., 2023), and approximately 15T tokens in the most recent models (Dubey et al., 2024).

Layer depth and d_{ffn} impact different types of benchmarks. We analyzed the key structural variables, including layer depth, d_{ffn}, d_{model}, and the number of attention heads, as shown in Figure 8a. The experiments conducted solely on the pre-trained model data can be found in Appendix C.1. Our findings indicate that the depth of layers primarily influences reasoning-related tasks, such as ARC-C, HellaSwag, and WinoGrande. In contrast, d_{ffn} has a greater impact on mathematical proficiency and knowledge accuracy, as observed in tasks like GSM8K, MMLU, and TruthfulQA.

This supports previous analyses: model depth determines the degree of non-linearity, enhancing reasoning capabilities (Jin et al., 2024; Mueller and Linzen, 2023; Ye et al., 2024), while empirical evidence suggests that LLMs store knowledge within the FFNs (Geva et al., 2020; Stolfo et al., 2023), where a larger d_{ffn} significantly improves memory capacity. It is also consistent with the findings that increasing the number of experts in MoE models, which can be viewed as an extension of feed-forward-neural layers, improves performance on knowledge-intensive tasks but not on reasoning tasks (Jelassi et al., 2024; Fedus et al., 2022).

Additionally, Mirzadeh et al. (2024) observes that even minor changes to the GSM8K dataset lead to a significant performance drop, suggesting that current LLMs have not yet mastered mathematical reasoning. At the same time, Stolfo et al. (2023) observes that large models primarily perform basic arithmetic operations in the feed-forward-neural layers. Both studies help explain why d_{ffn} is more important than the number of layers for the GSM8K dataset.

MMLU is the most representative benchmark. Our analysis shows that performance on MMLU is the most important feature for predicting model structure, as demonstrated by the computed feature importance values presented in Figure 8b. This supports the hypothesis that MMLU scores best reflect overall model performance. This finding aligns with the way organizations such as OpenAI, Anthropic, Mistral, and Qwen typically highlight 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 for pruning large language models, utilizing the Block Influence (BI) metric to assess the importance of different layers within the model. The BI score for the i^{th} layer is defined as:

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

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

By calculating the average BI scores across multiple benchmarks for the LLaMA-2-7B model, consistent trends emerge in the BI scores of various layers, as shown in Appendix C.2. As a result, using BI scores to analyze the distinct roles of each layer across different tasks is not straightforward. Additionally, we observed that the BI scores are generally higher in the earlier and last layers, while the scores in the middle-to-later layers tend to be lower, which aligns with the findings in (Men et al., 2024; Kim et al., 2024). Finally, layers 21 through 29, which exhibit the lowest BI scores, are selected for pruning in the experiment.

An anomaly is observed in the GSM8K benchmark, which requires models to generate precise numerical answers, unlike other benchmarks that typically involve selecting the most likely option from multiple choices. This difference in task structure introduces unique challenges for GSM8K, making

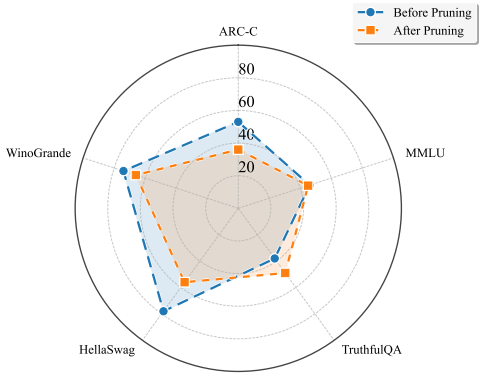


Figure 9: Performance across different benchmarks of Llama–2-7b before and after pruning 21-29 layers.

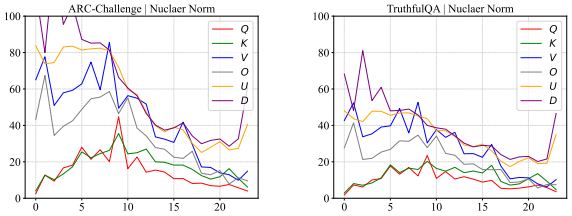


Figure 10: Layer-wise gradient analysis during fine-tuning of Qwen2 0.5B on the ARC_C and TruthfulQA benchmarks.

it less directly comparable to the other datasets. To ensure a fair evaluation across all benchmarks, we exclude GSM8K from this experiment.

Following the pruning of these layers, we evaluate the pruned model using the lm-evaluation-harness (Gao et al., 2024), with evaluation methods consistent with *the leaderboard*, and compare its performance before and after pruning across multiple benchmarks. The result is shown in Figure 9.

Pruning significantly affects performance on benchmarks where layer depth has the greatest impact (e.g., ARC-C, HellaSwag, WinoGrande), as shown in Figure 8a. In contrast, benchmarks with lower layer depth importance (e.g., MMLU, TruthfulQA) exhibit minimal degradation, with TruthfulQA even showing a slight improvement.

6.2 Validating Findings through Layer-wise Gradient Analysis

Following the gradient analysis methodology of Li et al. (2024), we evaluated the gradients when fine-tuning the Qwen-2-0.5B on the ARC-C and TruthfulQA benchmarks. Our validating targets includes the six major weight matrixes of each decoder layer, including Query (Q), Key (K), Value (V), and Output (O) projection in attention modules, and Up (U), and Down (D) projection in FFN modules. We denote $X \in \{Q, K, V, O, U, D\}$

The loss L_θ is aligned with the Cross-Entropy loss of next-token prediction used for supervised fine-tuning, where only responses contribute to the overall loss and instructions are ignored. We do multiple back-propagation until gradients from all entries of the dataset have been accumulated.

For the weight matrix of i-th layer X_i and it's corresponding gradient $G_{X,i}$, we measure the concentration of the gradient's spectrum on its dominant singular values through the Nuclear Norm $s_{X,i}$, witch offers insights into how the gradient behaves 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 $|\sigma_j|$ is the j-th singular value, computed with singular value decomposition algorithm, 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 the gradients in the later layers of the ARC-C dataset remain relatively high, indicating that deeper layers play a more crucial role in effectively completing the task. Which is aligned with our previous findings, that resoning tasks relies on the depth of the model. In contrast, the gradients in the deeper layers of the TruthfulQA dataset are comparatively lower, suggesting that these layers are not corresponding to memory tasks. A deeper look into gradients dynamic also proves our theory, given in Appendix C.3.

7 Conclusion

This research presents a comprehensive analysis of large language models (LLMs), introducing a large-scale dataset that captures various structure and performance metrics across multiple benchmarks. Through systematic examination, we trace the historical development of LLMs and discuss potential future trends. Our findings underscore the significant impact of structural configurations on model performance, with mechanistic interpretability techniques validating these discoveries. The study provides valuable, data-driven insights for optimizing LLM design, contributing to the development of more efficient, scalable, and adaptable models for a variety of applications.

Limitations

This study focused on specific datasets and tasks, which may limit the generalizability of our findings. Different applications may have distinct requirements and data characteristics. For example, instruction-following and coding tasks differ significantly from the datasets used in our analysis. Future work should involve testing our methods on a broader range of tasks and datasets to enhance the applicability of our results.

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

Ethics Statement

All training and evaluation datasets used in this research are publicly available and distributed under open-access licenses, intended solely for research purposes. These datasets do not contain any personal or identifying information, and no offensive content is included. The data collected in this study pertains to the model's structure and performance metrics.

All code and datasets used or created for this research are open-sourced under the MIT License. These resources are shared with the research community for further study and development, promoting transparency and reproducibility in research. We encourage other researchers to build upon and improve the resources provided, while adhering 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 LLM Structure and Performance Dataset

A.1 Detailed Description of Each Column

As shown in Table 3, 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 varient.
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.
likes	Like Count	1	The like count.

Table 3: Description of each column.

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

As shown in Table 4, the structure parameters of several models and their performance across different benchmarks are presented, including Llama-3-8B (Dubey et al., 2024), Bloom (Le Scao et al., 2023), Mixtral-8x7B, Llama-2-7B, and Mistral-7B (Jiang et al., 2023).

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

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	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 4: Examples from our LLM Structure and Performance Dataset.

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. 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 5.

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 5: Regression models and their key hyperparameters.

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

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: 55.20, HellaSwag: 78.22, MMLU: 50.30, TruthfulQA: 57.08, Winogrande: 73.24, GSM8K: 11.45
Answer1:
Prediction: ARC: 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: 41.38, HellaSwag: 70.26, MMLU: 25.63, TruthfulQA: 33.00, Winogrande: 66.46, GSM8K: 1.44
Answer2:
Prediction: ARC: 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 Regression Analysis of Pre-trained Model Structure and Performance

We conducted further experiments, performing regression analysis on data from models that had undergone only pretraining. A significant drop in the R^2 scores was observed for regression on MMLU, TruthfulQA, and GSM8K. Additionally, for predictions on datasets other than TruthfulQA, the importance shifted towards layers being more critical. This may be because the pre-trained model has not acquired specific domain knowledge for downstream tasks, leading to a lower significance of d_{ffn}. Furthermore, the most important feature for predicting model structure is not MMLU, but rather ARC-C and WinoGrande.

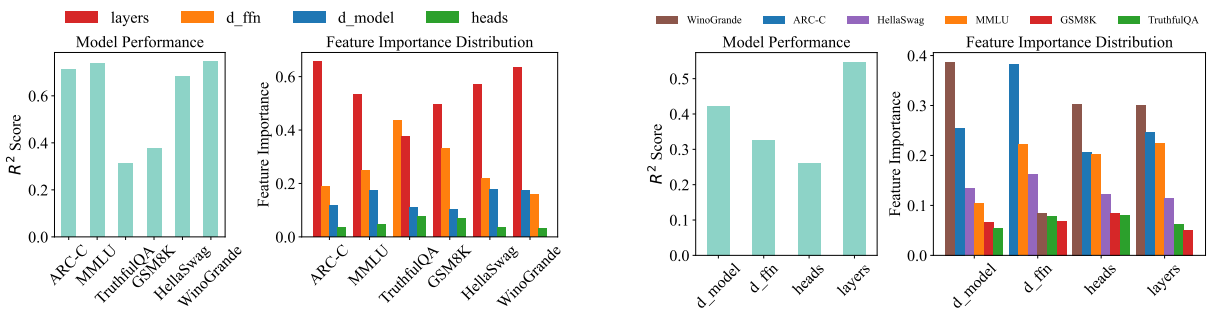


Figure 12: Regression analysis of pre-trained model structure and performance across benchmarks using Random Forest algorithm. (a) Predicting performance using structure; (b) Predicting structure using performance.

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

As shown in Figure 13, 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.

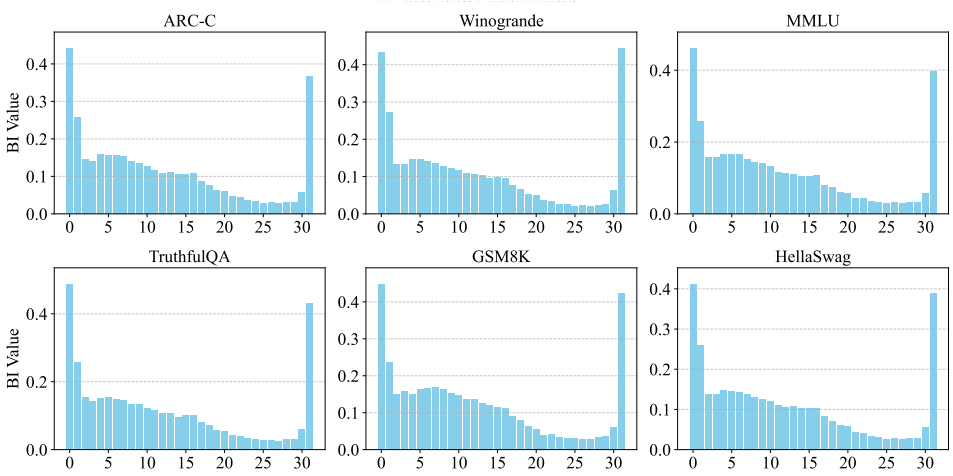


Figure 13: BI Scores of Different Layers in the LLaMA-2-7B Model Across Various Benchmarks.

C.3 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 14, 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.

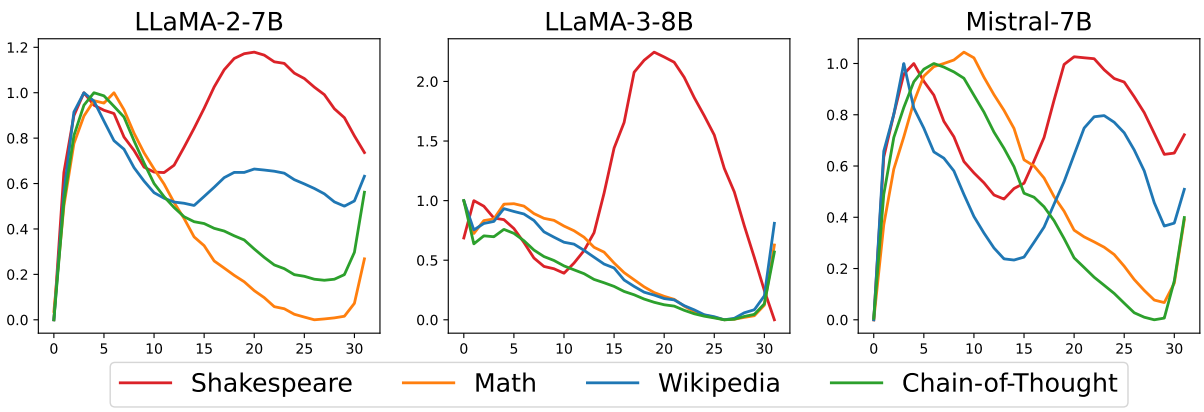


Figure 14: Layer-wise gradient on different corpuses.