Unveiling Biomedical Question Answering Pathways in Large Language Models: An Interpretability Study with Circuit Discovery

Xilin Wang

A senior thesis submitted in partial fulfillment of the

requirements for the Honors Degree

Department of Computer Science, Brown University
2025

Brown University

Abstract

Understanding the inner workings of large language models (LLMs) is essential for ensuring transparency in high-stakes domains such as biomedicine, where model decisions can directly impact patient outcomes. In this study, we apply mechanistic interpretability techniques to uncover and analyze computational circuits within LLMs during biomedical question answering. We employ Automatic Circuit Discovery (ACDC) in combination with Edge Attribution Patching (EAP) to enable scalable circuit extraction beyond toy settings. To facilitate comparative analysis, we finetune the base model and examine the differences in circuit structure before and after finetuning. Our analysis spans three levels of circuit granularity, revealing consistent model components and local motifs across model versions. We also observe structural changes induced by finetuning that lead to more efficient model computations. Overall, this work supports ongoing efforts toward transparent and interpretable LLMs in biomedicine through developing mechanistic understandings of model computations.

Acknowledgements

I would like to express my great gratitude to my advisor, Ritambhara Singh, who guided me throughout the entire year. In the beginning, the project did not go very well and I felt very lost, but Ritambhara has continuously worked with me and provided feedback that led to the completion of my thesis. This was a very meaningful research experience for me, and thank you so much for being there all the time.

I would also like to thank Michal Golovanevsky who shared this amazing idea of mechanistic interpretability for biomedical and clinical research. It was great to have your advice and guidance at the start of the project when I was unfamiliar with doing research and coming up with ideas.

Finally, I want to thank my family and my friends who supported me every day in my undergraduate life. It was an honor to have you on my back all the time.

Contents

Chapter 1

Introduction

With the recent developments of Large Language Models (LLM) and their successful applications in a wide range of natural language processing tasks, people's attention has been shifted to domain-specific LLMs adapted to domain-specific corpora to outperform general-purpose LLMs[14]. Many biomedical LLMs are developed to process biomedical text data such as the unstructured texts in the Electronic Health Records (EHR)[8]. Subsequently, several benchmarks for biomedical question answering emerges, including MedQA, PubMedQA, etc.[6; 7]. A rising concern for using LLMs in more high-stake applications lies in the black-box nature of these models; This is particularly problematic in the biomedical and clinical domain, where model decisions directly influence disease diagnosis, assessment plans, and medications, further affecting patient outcomes. Subsequently, promoting the interpretability of LLMs in the biomedical and clinical domain is necessary before their more extensive applications in scientific discovery and clinical guidance.

LLM interpretation is defined as *the extraction of relevant knowledge from an LLM, either inherently learned by the model or contained in the data* [14]. Methods such as attention weights and self-explanatory instruction have been applied to understand the biomedical and clinical capabilities of LLM [5; 13]. On the other hand, mechanistic interpretability has emerged as a separate line of research that aims to reverse-engineer model computations into explicit, human-understandable formats [1; 3]. To the best of our knowledge, there has been little work done on explaining LLM's biomedical capabilities using these global explanation approaches.

This paper aims to apply mechanistic interpretability methods to identify and compare circuits from LLMs corresponding to computations during biomedical question answering. This goal directly aligns with the two main challenges of Explainable Artificial Intelligence (XAI) proposed by Renftle et al., approximation and translation: 1) Approximate complex LLMs using simpler circuit representations and 2) Translate technical attributes, such as attention

heads and their causal relationships with inputs, into more interpretable patterns such as unique motifs for biomedical models compared to general purpose models [12].

We follow the pipeline Conmy et al. summarize for identifying a circuit, particularly the Automated Circuit Discovery (ACDC) algorithm that iteratively patches the model in an automatic manner[3]. Our corrupted dataset removes the biomedical contexts during question answering, thereby recovering model components responsible for biomedical knowledge comprehension from the inputs during model generation. We additionally incorporate Edge Attribution Patching (EAP) to speed up computations using gradient-based approximations [15]. To better interpret the circuits, we fine-tune our base model, Llama-3.2-1b-instruct, on the training partitions of PubMedQA and compared circuits recovered from the original model and its adapted version.

We show that fine-tuning LLMs on PubMedQA improves the model's performance in terms of both accuracy and low-level logit differences. Through comparative circuit analysis, we find that model components like MLP layers have consistent appearance in computations of biomedical question answering, while attention heads exhibit variability across circuits of different models. Our circuits share local motifs in which same attention heads are attached to particular MLP layers. At each scale, the circuit of the fine-tuned model are more connected, with a distributional shift towards earlier layers, reflecting an improved efficiency of computation on biomedical question answering. By offering mechanistic insights into model architectures, our findings aim to promote LLM transparency in biomedical applications.

Chapter 2

Related Work

Prior works on LLM interpretation can be broadly categorized as either explaining a single input-output generation from the LLM, or explaining LLM in its entirety. One of the most popular approaches to examine a one-take generation is assigning feature importance scores, i.e. identifying most predictive parts of the inputs to the model generation. For example, previous work used attention-based methods to interpret LLMs, calculating input importance scores based on the attention weights for corresponding input features[9]. Huang et al. developed the ClinicalBERT model and extracted the attention weight distribution for each query vector to visualize terms in clinical notes that are important for model prediction[5]. Other studies applied Shapley additive explanation (SHAP) to train linear models that represent important combinations of features at the expense of exponentially increasing computational costs[9]. Recent work also explores the potential of prompting LLMs to explain themselves, such as chain-of-thought prompting where models are instructed to output the intermediate reasoning process before arriving at the final answer[17]. Savage et al. demonstrated the use of diagnostic reasoning prompts that encourage LLMs to imitate clinical reasoning processes[13]. However, such methods heavily rely on language models to interpret themselves, leaving the space for model hallucination and thus unreliable explanations.

Global explanation goes beyond individual input-outputs and focuses on mechanistic understandings of how models function based on their internal architectures[14]. Methods in this category include probing model components like attention heads and embeddings and understanding how several model components collectively perform specific tasks[2; 16]. The results from mechanistic interpretability methods are usually much simpler algorithms, or circuits, accountable for specific computations. For example, Wang et al. discovered several classes of attention heads that collaboratively form a circuit that performs the indirect object identification (IOI)[16]. Compared to traditional interpretation methods, mechanistic interpretability is more granular as it examines the inner workings of the model

architecture. However, given the capacity of LLMs, it is difficult for this line of research to scale up beyond toy tasks and models in natural language processing despite few attempts of circuit analysis at scale[19]. This paper attempts to moderately scale up circuit discovery with the particular focus on biomedical question answering, aiming to recover important model components and promote trustworthiness of model predictions in the biomedical fields.

Chapter 3

Methods

3.1 Mechanistic Interpretability

We apply the workflow of mechanistic interpretability introduced by Conmy et al.[3]. The goal is to visualize and understand model computations during biomedical question answering through recovering the *circuits*, formally defined as follows (Wang et al.): We represent a model, in this scenario an LLM, as a computational graph, where nodes are individual model components (attention heads, individual neurons, etc.) and edges are connections among these components (projections, residual connections, etc.). During a forward pass, a set of inputs goes through this computational graph to produce an output. In this process, certain nodes and edges are more responsible than other components for the output generation. A circuit is subsequently a subgraph of the whole computational graph that represents model computations on the particular task [16].

The first step is to select a specific task (dataset and metric) that displays the desired behaviors[3]. PubMedQA is a dataset for biomedical question answering composing of research abstracts from PubMed[7]. A model is instructed to answer the research question with yes/no/maybe using the corresponding abstract. This dataset evaluates a model's ability to draw from its internal knowledge base as well as relevant knowledge from the abstracts to answer biomedical questions. Based on the categorical outputs, we choose the logit difference between correct and incorrect answers as our metric to measure model performance, where larger logit difference reflects more confident prediction towards the correct answer. Notably, this task is more difficult than natural language processing tasks used in previous work such as Indirect Object Identification (IOI), representing our effort to scale up computations beyond toy tasks.

The second step is to define the scale at which circuits are recovered[3]. We select the level of abstraction for the computational graphs to be individual attention heads and Multilayer Perceptron (MLP) layers in accordance with the

```
Provide a single word answer to the
    following research question based
    on the research abstract.
###Abstract:
###BACKGROUND: Programmed cell death
    (PCD) is the regulated death of
    ...
(background continues)
###RESULTS: The following paper
    elucidates the role of
    mitochondrial dynamics ...
(results continue)
###Question: Do mitochondria play a
    role in remodelling lace plant
    leaves during programmed cell
    death?
Directly answer yes/no:
```

```
Provide a single word answer to the
    following research question based
    on the research abstract.
###Question: Do mitochondria play a
    role in remodelling lace plant
    leaves during programmed cell
    death?
Directly answer yes/no:
```

Figure 3.1: Side-by-side comparison of the original sample (left) and the corrupted sample (right)

goal of analyzing and comparing circuits responsible for biomedical question answering. This granularity ensures the discovery of important attention heads and MLP layers without going too deep into individual neurons.

The last step is to search for important nodes and edges that collectively form the circuit [3]. We use *activation patching* to iteratively remove as many unimportant model components as possible. Activation patching refers to the process of overwriting the activation of a model component with a corrupted value and comparing the results from forward passes with and without this corruption based on the predefined metric. While several work performs patching manually [16], we utilize an algorithm termed Automatic Circuit Discovery (ACDC, Section 3.2) to automate this process for better scaling [3].

Activation patching crucially relies on the definition of corruption. Some works have overwritten nodes with zero vectors[11], while others have chosen mean activation values across a reference dataset as an uninformative input[16]. In the specific context of PubMedQA, prompts do not follow structurally similar patterns as observed in toy tasks since different research questions are formulated differently. Therefore, we select interchange interventions introduced by Geiger et al., which replace activation values of the original data point with those of a closely related but uninformative data point [4]. For each sample in PubMedQA, we remove the corresponding research abstract and construct the corrupted sample (Fig. 3.1). This corruption erases information that helps LLMs answer the research question and forces the model to use its internal knowledge base accumulated during pretraining. Our choice of corruption schemes is grounded in the belief that it is more complex and obscured to visualize model memories; instead, we recover circuits that are responsible for understanding and leveraging relevant biomedical knowledge within the prompt.

3.2 Automatic Circuit Discovery

The core algorithm that performs activation patching automatically is described in Fig. 3.2 [3]. The algorithm iteratively goes through the whole computational graph *G* in reverse order, removing as many incoming edges as possible at every node. Edge removal in this case refers to the action of overwriting the incoming node value under the original dataset, $(x_i)_{i=1}^n$, with the ablated value calculated using the corrupted dataset, $(x_i')_{i=1}^n$, for each data point. We replace KL-divergence with logit difference in the algorithm and calculated the mean logit differences across n data points. Finally, the change in metric is compared to a predefined thresholdτ: we remove the edge if this action does not lead to a large decrease in the mean logit difference.

Algorithm 1: The ACDC algorithm.

Data: Computational graph G, dataset $(x_i)_{i=1}^n$, corrupted datapoints $(x_i')_{i=1}^n$ and threshold $\tau > 0$.
Result: Subgraph $H \subseteq G$.

```
H ← G                                  // Initialize H to the full computational graph
H ← H.reverse_topological_sort()       // Sort H so output first
for v ∈ H do
    for w parent of v do
        H_new ← H \ {w → v}            // Temporarily remove candidate edge
        if D_KL(G||H_new) − D_KL(G||H) < τ then
            H ← H_new                  // Edge is unimportant, remove permanently
        end
    end
end
return H
```

Figure 3.2: Automatic Circuit Discovery Algorithm

Our experiments successfully recovered circuits using toy models such as GPT-2 small. Nonetheless, activation patching is computationally expensive as each patching step requires another forward pass to compute new metric, resulting in computational overheads as the model parameters grow beyond several hundreds of millions. Following Syed et al., We describe a new methodology built on top of ACDC that performs linear approximation to directly estimate importance scores for each edge in the computational graph, termed Edge Attribution Patching (EAP, Section 3.3) [15].

3.3 Edge Attribution Patching

The major computational overhead of activation patching is the forward passes per each edge removal during the iterative patching process. Define $L(H)$ be the metric, in this case logit difference, computed with the current model subgraph H and input x. We want to evaluate the change in metric for each edge e:

$$L(H/e) - L(H) \tag{3.1}$$

H/e represents the temporary model subgraph with edge e removed. Rather than directly calculating this value through another forward pass, we use the first-order gradient-based approximation (Equation 3.2).

$$L(H/e) - L(H) \quad \approx \quad (e_{\text{corr}} - e_{\text{clean}})^\top \frac{\partial}{\partial e_{\text{clean}}} L(H) \tag{3.2}$$

Notably, computing EAP scores only requires two forward passes and one backward passes on the dataset, largely reducing the time complexity issue in activation patching [15]. After the computation, thresholding is again applied to select the top k important edges based on the EAP scores. In this work we directly filter out top k edges with varying k rather than specifying a score threshold to easily manipulate circuit sizes.

3.4 Model Fine-tuning

After recovering circuits, we analyze the model components and hypothesize their respective functions during computation [3]. However, circuits are usually hard to interpret from a first glance. To better understand these graphs, we propose to perform fine-tuning on the model and extract circuits at the same scale, thereby obtaining pairs of circuits for downstream analysis.

Our model fine-tuning is similar to the instruction tuning phase proposed by Wu et al. [18]. However, while their goal is to create a lightweight medical foundation model (PMC-LLaMA), we aim to directly improve model performance on PubMedQA. Therefore, our fine-tuning is built on top of the instruction-tuned versions of open-sourced foundation models instead of their pre-trained counterparts to better utilize their instruction-following capabilities. We follows the standard instruction-tuning objective, where the next-token-prediction loss on model responses is minimized. We do not use the loss with respect to the instructions provided to the model.

Chapter 4

Results

4.1 Experiment Setup

4.1.1 Dataset

The pqa-labeled partition of PubMedQA, consisting of 1k research questions, is used as the dataset for circuit discovery. We disregard the rows where the final answer is "maybe" for more straightforward and stable performance evaluation. We also limit the token lengths of the dataset to be under 400 due to the space complexity EAP imposes, where longer prompts consume more memories during forward and backward passes. Finally, 100 samples are randomly selected as the dataset on which EAP is performed. We padded the input tokens using the maximum length from the dataset before feeding into the model.

Our instruction tuning dataset consists of 211.3k artificially generated samples included in PubMedQA's pqa-artificial subset, following the same format as question answering pairs [18].

4.1.2 Model Selection

We select Llama-3.2-1B-Instruct as our base model. Although this model does not reach the scale of industrial-level foundation models, its 1 billion parameters is approximately ten times those of GPT 2-small used by many mechanistic interpretability research. The model is loaded in with float16 precision to again reduce memory overhead in EAP, and same precision is used during fine-tuning.

4.1.3 EAP Specification

During EAP, we only consider attention heads and MLP layers in the model and disregard the residual streams. There are two considerations: 1) EAP performs badly on "big" activations like an entire residual stream [10] 2) including

```
Below is an instruction that describes a
    task, paired with an input that
    provides further context. Write a
    response that appropriately completes
    the request.\n\n
###Instruction:\n{instruction}\n\n
###Input:\n{input}\n\n
###Response:
```

Figure 4.1: Instruction Tuning Prompt

residual streams empirically increases memory requirements.

4.1.4 Training Details

We fine-tune Llama 3.2 1b instruct for 3 epochs with a batch size of 256, resulting in a total of 2463 training steps using LoRA. We use AdamW optimizer with a learning rate of $2e-5$ coupled with linear LR scheduler. Our prompting strategy follows Wu et al. illustrated in Figure 4.1 [18]. We remove any special tokens that proceed the actual answers to make sure that the first generated logit corresponds to the model's actual response to the research question.

4.2 Model Performance

We first present the fine-tuning results. Both the original Llama 3.2 model and the fine-tuned version are instructed using the testing set of PubMedQA. In this case we do not limit the token length to fully observe improvements due to fine-tuning. Table 4.1 compares different metrics calculated using llama 3.2 and the fine-tuned version, respectively. Models that are labeled as "corrupted" refer to the results using the corrupted dataset where research abstracts are removed. Among the four experiments, the fine-tuned Llama 3.2 achieved the best overall accuracy of 0.67 and the best Micro F1 score of 0.79. On the other hand, the untuned Llama 3.2 achieved the best Macro F1 at 0.63. In either case, corrupting the dataset by removing research abstracts negatively impacts the model's performance. Compared to the original model, our fine-tuned model suffered less from corrupted data based on the metric differences.

Table 4.1: Fine-tuning Results on PubMedQA

Metric	Llama 3.2	Llama 3.2 (corrupted)	Llama 3.2 Fine-tuned	Llama 3.2 Fine-tuned (corrupted)
Accuracy	0.63	0.48	**0.67**	0.64
Micro Precision	**0.79**	0.67	0.65	0.64
Micro Recall	0.55	0.34	**0.99**	0.97
Micro F1	0.65	0.45	**0.79**	0.77
Macro Precision	0.65	0.53	**0.77**	0.66
Macro Recall	**0.66**	0.53	0.56	0.54
Macro F1	**0.63**	0.48	0.51	0.49

In addition to the above metrics, we calculate logit differences between correct and incorrect answers for respective models, which align with the metric used in circuit discovery (Table 4.2). Fine-tuning improves both clean and corrupted logit differences, with a larger change observed in the former. This indicates that our model not only absorbs biomedical knowledge from the dataset, but also learns to leverage the research abstracts from the input to produce correct answers.

Table 4.2: Logit Differences

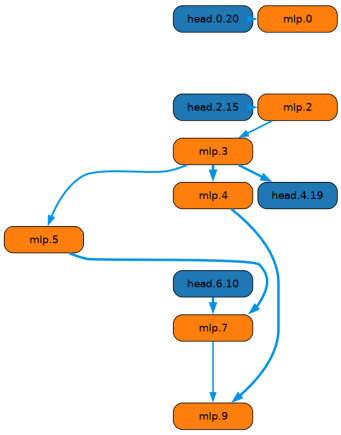
Metric	Llama 3.2	Llama 3.2 Fine-tuned
Clean Logit Difference	0.61	2.17
Corrupted Logit Difference	0.38	1.44

4.3 Circuits

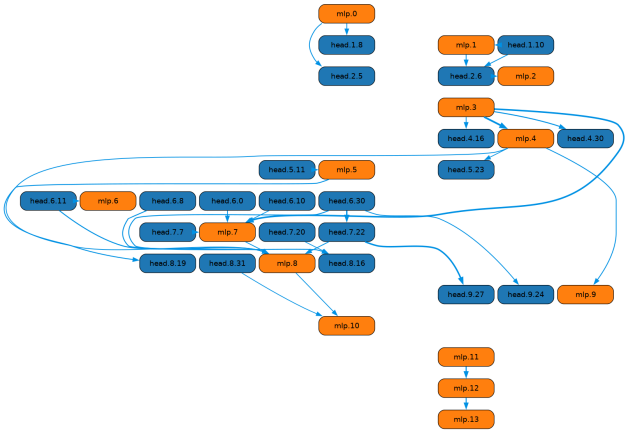
This section visualizes circuits obtained from EAP at three different scales, depending on the choices of the number of edges. For each scale, we present the circuits recovered from the original Llama 3.2 1b instruct model and our fine-tuned model. We use $k = 10$, 32, and 100 to recover circuits with a small, medium, and large scale, respectively. Note that the specification of k essentially adds a threshold to the list of edges ordered in terms of their importance. Therefore, a large-scale circuit includes most, if not all considering small numerical instability during forward passes, edges from the corresponding smaller-scale circuit plus additional edges that are less important.

Figure 4.2 presents our recovered circuits. In each graph, the thickness of an edge represents its importance score. Since the resulting importance scores are much larger for the original model, we scale the numerical values for edges in the fine-tuned circuits to be 5 times larger during visualization. This process does not affect the recovered circuits as the relative edge importances remain the same.

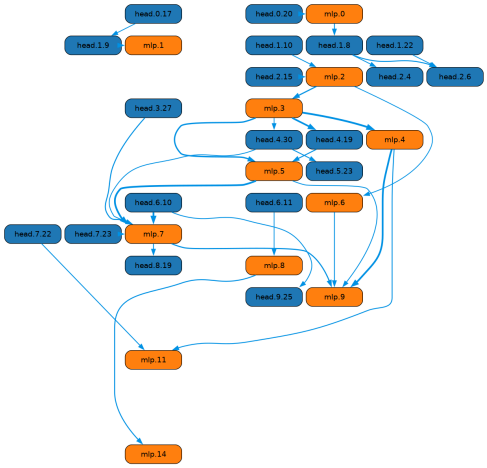
(a) Original model, k=10



(b) Fine-tuned model, k=10



(c) Original model, k=32



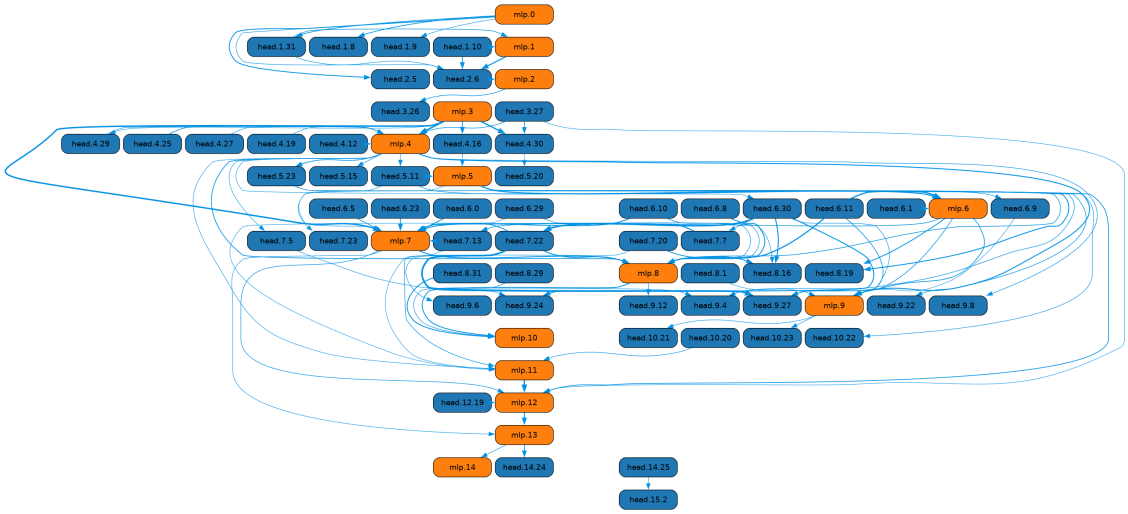
(d) Fine-tuned model, k=32

Figure 4.2: Recovered circuits at small and medium scales. Within each scale, the first panel shows the original Llama 3.2 1b instruct model and the second panel our fine-tuned counterpart.

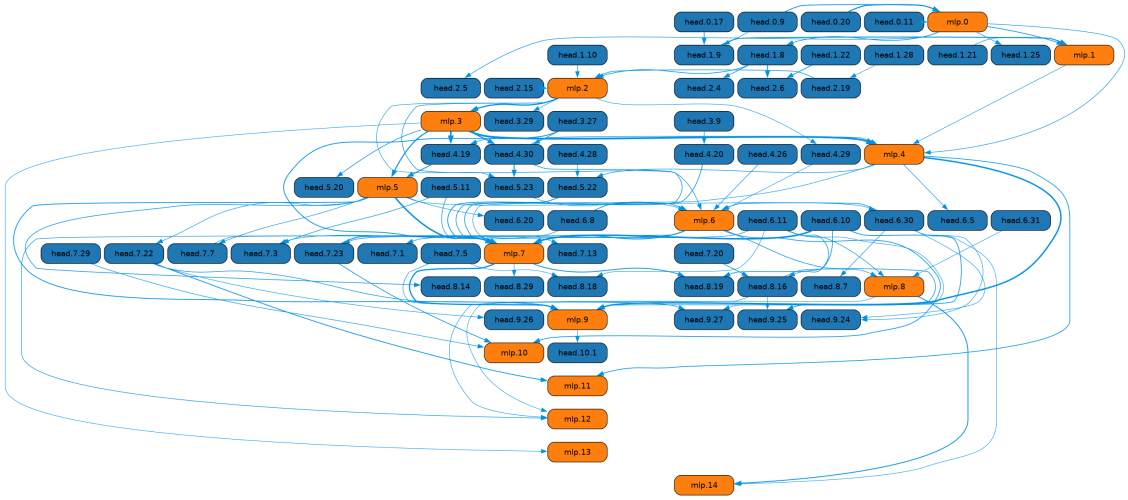
In Figures 4.2a and 4.2b, nodes like $mlp.3$, $mlp.4$, $mlp.5$, and $mlp.7$ consistently appear in both circuits. By contrast, the use of attention heads seem to vary much more in two models. Circuits of our fine-tuned model appear more connected than circuits of the original model. For example, Figure 4.2c has four separate groups of nodes, while Figure 4.2d only has two groups with a major group consisting of most nodes in the graph. Although both models are connected in an end-to-end fashion in terms of their architectures, a more connected circuit shows that model computations are more coherent and centralized, with task-specific model components activated closely with each

other.

Figures 4.2e and 4.2f are large-scale circuits with $k = 100$. 17 out of 100 edges are shared in the two circuits. All MLP layers except $mlp.15$ are consistently present in both circuits, demonstrating that major model computations happen both at early layers and at later ones when using a lower threshold. There are 55 and 56 attention-head nodes in the original model's circuit and the fine-tuned model's circuit, respectively, among which 28 attention heads are shared.



(e) Original model, k=100



(f) Fine-tuned model, k=100

Figure 4.2: (continued) Circuits at large scale.

Chapter 5

Discussion

5.1 Fine-tuning

Our work relies on LLM fine-tuning to obtain a task-specific counterpart of the base model and to establish the basis for comparative circuit analysis. We found that fine-tuning improves the model's performance on biomedical question answering when measured through accruacy and Micro F1; however, it yields a lower Macro F1 compared to the original model. One reason is that the fine-tuning dataset is unbalanced towards positive samples, resulting in a tendency for the model to produce "yes" even when it is unsure of the correct answer. This results in a very high Micro recall but a low Micro precision. The model achieves a relatively high Macro precision due to a high precision for negative samples; simultaneously, it suffers from low recall for negative samples.

5.2 Comparative Circuit Analysis

Across different models and scales, MLP nodes appear to be relatively stable, indicating that they play fundamental roles in biomedical question answering, possibly capturing critical biomedical knowledge and semantic patterns. By contrast, attention heads exhibit greater flexibility especially in small to medium scale circuits. In particular, fine-tuning may affect how models originally take care of the input contexts, assigning tasks more efficiently to new sets of attention heads. This hints at a possible separation of labor within the model, where MLPs serve as stable feature extractors and attention heads adapt dynamically to task-specific needs.

Fine-tuning significantly alters the circuit structures beyond mere weight updates. Circuits from the fine-tuned model are more densely connected, suggesting a more centralized use of nodes that specializes for biomedical question answering. Additionally, major computations seem to finish earlier for fine-tuned circuits in which few nodes after

layer 7 appear (Figure 4.2c and 4.2d). Nodes such as $mlp.9$ might serve as the destination for knowledge transfer with its incoming edges, while the bulk of computation is completed in earlier layers. This indicates that fine-tuning possibly promotes a more efficient computational pathway for performing the task.

In large scale circuits ($k = 100$), we observe that although a large number of attention heads differ between the original and fine-tuned models, there remains substantial overlap (28 shared attention heads and nearly all shared MLP layers). For example, $mlp.7$ in the two circuits share several consistent edges, including incoming edges from $head5.11$, 6.10, and 7.7. While attention patterns adapt during task-specific fine-tuning, core computation backbones may be anchored by MLP layers and are reflected as consistent motifs shared among circuits.

5.3 Limitations and Future Work

We successfully recovered circuits from llama 3.2 consisting of 1 billion parameters, substantially larger than the models traditionally analyzed with ACDC. This underscores the value of EAP in scaling mechanistic interpretability beyond toy tasks to LLMs used in practice. However, Llama-3.2-1b-Instruct remains much smaller than state-of-the-art biomedical LLMs. Scaling EAP to models with tens of billions of parameters remains computationally prohibitive.

While we carefully designed our corruption strategy to focus on the model's capability of biomedical context usage, the corruption might not fully isolate internal memory usage. Future work that involves more sophisticated corruption schemes such as perturbations of important biomedical entities may lead to circuits that are more aligned with specific biomedical reasoning patterns.

Our methodology qualitatively analyzes properties such as model components and circuit density. Further work might look into detailed and interpretable functions of individual nodes, annotating recovered circuits by probing individual node behaviors through causal tracing techniques.

Previous work showed that attribution patching works poorly on large activations such as residual streams, as gradient-based methods might produce inaccurate approximations on big edges [10]. Future work could consider using activation patching (ACDC) on top of EAP to recover more truthful circuits. Additionally, it might be interesting to observe how circuits evolve during different stages of fine-tuning (early, mid, late), and we leave this exploration for future work.

Chapter 6

Conclusion

In this work, we demonstrated the feasibility and potential of applying mechanistic interpretability techniques, particularly Automatic Circuit Discovery (ACDC) and Edge Attribution Patching (EAP), to analyze LLM computations in biomedical question answering. Our analysis revealed that certain model components in the form of local motifs consistently play crucial roles across different models, suggesting their fundamental importance in biomedical question answering. The increased connectivity and the accelerated computation observed in the fine-tuned model's circuits indicate a more efficient utilization of model components, potentially contributing to improved performance and robustness. These findings underscore the potential of mechanistic interpretability methods to provide deeper insights into LLM behaviors, especially in high-stakes domains like biomedicine, bridging the gap between complex model architectures and human-understandable explanations.

Bibliography

[1] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for AI safety — a review. *Transactions on Machine Learning Research*, 2024. Survey, accepted at TMLR; arXiv:2404.14082.

[2] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, 2019. Association for Computational Linguistics.

[3] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS) 36*, 2023.

[4] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34 of *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, pages 9574–9586, 2021.

[5] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[6] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[7] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, 2019.

[8] Zhiyong Lu, Yifan Peng, Trevor Cohen, Marzyeh Ghassemi, Chunhua Weng, and Shubo Tian. Large language models in biomedicine and health: Current research landscape and future directions. *Journal of the American Medical Informatics Association*, 31(9):1801–1811, 2024.

[9] Daoming Lyu, Xingbo Wang, Yong Chen, and Fei Wang. Language model and its interpretability in biomedicine: A scoping review. *iScience*, 27(4):109334, 2024.

[10] Neel Nanda. Attribution patching: Activation patching at industrial scale. `https://www.neelnanda.io/mechanistic-interpretability/attribution-patching`, February 2023. Blog post.

[11] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[12] Moritz Renftle, Holger Trittenbach, Michael Poznic, and Reinhard Heil. What do algorithms explain? the issue of the goals and capabilities of explainable artificial intelligence (XAI). *Humanities and Social Sciences Communications*, 11:760, 2024.

[13] Thomas Savage, Ashwin Nayak, Robert Gallo, Ekanath Rangan, and Jonathan H. Chen. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine. *npj Digital Medicine*, 7(1):20, 2024.

[14] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024. cs.CL.

[15] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[16] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in*

Neural Information Processing Systems, volume 35 of *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, pages 24824–24837, 2022.

[18] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-llama: Towards building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843, 2024.

[19] Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah D. Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Advances in Neural Information Processing Systems*, volume 36 of *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, pages 78205–78226, 2023.