Mechanistic Analysis Of Universality: Numerical Comparison Circuits Across Transformer Architectures

Arya Bhardia[1], Julian Ramirez[2], Siddhanta Verma[3], Karen Mkrtchyan[4]

[1]University of California - Berkeley, [2]New York University,
[3]University of California - San Diego, [4]University of Southern California
[1]arya_bhardia@berkeley.edu, [2]jr7281@nyu.edu,
[3]syverma@ucsd.edu, [4]kmkrtchy@usc.edu

Abstract

Transformer language models reliably achieve high accuracy on many reasoning tasks; however, their internal mechanisms are not fully understood. Mechanistic interpretability seeks to remedy this gap by identifying task circuits within individual models, but it is unclear whether such circuits generalize across model families and scales. In this work, we study the universality of circuits through the lens of numerical comparisons, a simple and controlled task that enables clean and causal interventions. We conduct experiments on a set of transformer models spanning different families and sizes from 1.7b to 9b parameters. We find that models within the Qwen family exhibit a highly consistent circuit structure across architecture and scale, featuring localized attention heads that write a task relevant signal. In contrast, models from other families show qualitatively different implementations, where task relevant information emerges much earlier and is distributed across components as opposed to being concentrated within a small set of attention heads. These results serve as evidence that task behavior similarities do not imply mechanistic universality and highlight the necessity for cross model comparisons to claim generalization of internal circuits.[1]

1 Introduction

Transformer models have demonstrated broad capabilities, including emergent behaviors that arise primarily from scale rather than task-specific architectural changes (Brown et al., 2020; Wei et al., 2022). There has been a prevalent motivation to understand how these models implement computations internally beyond just purely behavioral evaluation. In this field of research, mechanistic interpretability aims to reverse engineer transformers by identifying internal representations that implement specific algorithmic behaviors (Elhage et al., 2021; Olah et al., 2022b,a). Recent work has also developed practical intervention based tools for localizing and validating such circuits (Heimersheim and Nanda, 2024; Wang et al., 2023). In the context of numerical reasoning, previous results have shown that small models can implement structured internal mechanisms for "greater than" tasks (Hanna et al., 2023), suggesting that numerical comparison behaviors may be driven by interpretable internal structures instead of blunt memorization and pattern matching.

A separate but related line of work questions whether these internal representations and circuits are generalizable across models: that is, whether neural networks, trained with different objectives on different data and modalities, have converging representations. Prior work on neural network representations suggests that different models are capable of having shared internal spaces despite differences in training dynamics (Kornblith et al., 2019). More recent hypotheses propose that high level features may be shared across architectures (Huh et al., 2024). As such, there has been growing interest in whether mechanistic circuits generalize across different settings (Wang et al., 2024; Räuker et al., 2023; Zhao et al., 2024; Ferrando and Costa-jussà, 2024). In this work, we investigate numerical comparison as a controlled setting for testing mechanistic universality, using causal interventions to localize greater-than computation and compare the extent to which analogous circuits emerge across multiple transformer families.

2 Setup

To identify adequate models, we constructed a 4-digit comparison prompt benchmark in the following format: "Is 1923 $>$ 1987? Answer:". This differs from other approaches such as "The war

[1]The code and data for this work is available at https://github.com/KarenMkrtchyan/Mechanistic_Analysis_of_Universality

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 951–967

Model	Accuracy (%)	Number of Layers	Number of Heads
Qwen3-1.7b	99.8	28	16
Qwen2.5-3b	99.8	36	16
Qwen3-4b	99.7	36	32
Qwen2.5-7b	99.7	28	28
Llama3-8b-instruct	98.1	32	32
Gemma2-9b-instruct	99.8	42	16

Table 1: Model benchmark results with architecture details (layers and heads) compiled from TransformerLens documentation (Nanda et al., 2023). Models include Qwen2.5-3b and Qwen2.5-7b (Qwen et al., 2025) Qwen3-1.7b and Qwen3-4b (Yang et al., 2025). Llama3-8b-instruct (Grattafiori et al., 2024), and Gemma2-9b-instruct (Team et al., 2024)

started from 1923 and ended in 19. . . .”(Hanna et al., 2023), but benefits us in finding a more algorithmic boolean circuit rather than induction-like behavior. Because the models we analyze do not tokenize whitespace individually at the end of our prompts, we focus on the output tokens “ Yes” and “ No”. The benchmark consists of 1000 randomly generated prompts and a prediction is considered correct if the model assigns a higher logit to the correct truth value token rather than the incorrect token. The models selected for analysis have achieved at least 98% accuracy on this benchmark.

For our subsequent experiments, using the same format as listed below, we create contrastive clean and corrupt prompt pairs by swapping the numbers. Clean prompts are designed to have truth values of “ Yes", while corrupt prompts are the opposite with truth values of “ No".
Clean - “Is XXXX $>$ YYYY? Answer:”.
Corrupt - “Is YYYY $>$ XXXX? Answer:”

3 Methodology

We first establish the notation for the transformer architecture. A transformer layer consists of an attention layer and a feedforward network block (FFN). We adopt the notation and architectural description from Ferrando et al. (2024), where at a decoding step i each attention reads from the residual streams at previous positions ($\leq i$). The attention block is composed of multiple attention heads, and each head computes

$$\mathrm{Attn}^{l,h}(X_{\leq i}^{l-1}) = \sum_{j \leq i} a_{i,j}^{l,h} x_j^{l-1} \mathbf{W}_{OV}^{l,h} \qquad (1)$$

where $\mathbf{W}_{OV}^{l,h}$ is the combined value and output matrices, $x_{\leq i}^{l-1}$ is the residual stream at the previous layer, and $a_{i,j}^{l,h}$ are the attention weights.

The attention block output is the sum of individual attention heads, which is subsequently added back to the residual stream.

$$\mathrm{Attn}^{l}(X_{\leq i}^{l-1}) = \sum_{h=1}^{H} \mathrm{Attn}^{l,h}(X_{\leq i}^{l-1}), \qquad (2)$$

$$x_i^{mid,l} = x_i^{l-1} + \mathrm{Attn}^{l}(X_{\leq i}^{l-1}) \qquad (3)$$

The FFN block is composed of two learnable weight matrices: $W_{in}^{l} \in \mathbb{R}^{d \times d_{ffn}}$ and $W_{out}^{l} \in \mathbb{R}^{d_{ffn} \times d}$. W_{in}^{l} reads from the residual stream state $x_i^{mid,l}$, and its result is passed through an element-wise non-linear activation function g, producing the neuron activations. These get transformed by W_{out}^{l} to produce the output $\mathrm{FFN}(x_i^{mid})$, which is then added back to the residual stream.

$$\mathrm{FFN}^{l}(x_i^{mid,l}) = g(x_i^{mid,l} \mathbf{W}_{in}^{l}) W_{out}^{l} \qquad (4)$$

A mathematical framework formalizes circuit discovery as tracing information flow through transformers, linking representation transforms with emergent behaviors (Elhage et al., 2021). We begin looking for a circuit in Qwen2.5-3b to solve the numerical comparison task by focusing on three complementary approaches: activation patching, path patching, and direct logit attribution (All experiments were implemented using the TransformerLens library (Nanda et al., 2023). Each method provides a way of quantifying the causal or functional role of a component in producing the correct output token.

3.1 Activation Patching

Activation patching (Heimersheim and Nanda, 2024) is a causal intervention technique that we use to identify which internal components of the transformer are responsible for promoting the correct token output. Let x_{clean} and x_{corrupt} be clean and corrupt prompts, respectively, and let $h_\ell(x)$ denote the intermediate activations of some component in the layer ℓ. During a forward pass on a corrupt

input, we can intervene on a model component by replacing the intermediate activations produced by the corrupt input $h_\ell(x_{\text{corrupt}})$ with the corresponding activations from the clean input $h_\ell(x_{\text{clean}})$. We then measure how much the prediction changes between the two runs by recording a normalized logit difference between the correct and incorrect answer tokens

$$\Delta z_t = \frac{z_t(x_{\text{patched}}) - z_t(x_{\text{corrupt}})}{z_t(x_{\text{clean}}) - z_t(x_{\text{corrupt}})}. \quad (5)$$

where $z_t(x)$ is the logit of the token t given input x. Normalizing the equation ensures that values are comparable across prompts. If patching a component shifts the model's prediction on the corrupt prompt toward the clean prompt's answer token (indicated by a positive Δz_t), then the component may play a causal role in the task.

3.2 Direct Logit Attribution (DLA)

Direct Logit Attribution (DLA) is a decomposition technique to measure the contribution of individual internal components to the final logits of specific output tokens by rearranging the traditional forward pass formulation as shown in Equation (6).

$$\left(\sum_{l=1}^{L} \sum_{h=1}^{H} \text{Attn}^{l,h}(\mathbf{X}_{\leq n}^{l-1}) + \sum_{l=1}^{L} \text{FFN}^{l}(\mathbf{x}_n^{\text{mid},l}) + \mathbf{x}_n \right) \mathbf{W}_U \quad (6)$$

This allows us to measure the direct contribution of every model component to the logits of the predicted token (Ferrando et al., 2024). Given a target token t we can record the contribution of a vector, h_ℓ, to the logit of t by projecting its output onto the unembedding matrix $\mathbf{W}_U$ to compute its direct influence.

$$\text{DLA}(h_\ell, t) = h_\ell^\top W_U[:, t] \quad (7)$$

This provides a means by which we can evaluate how much a component directly increases or decreases the logit of a specific candidate token.

3.3 Path Patching

Path patching (Wang et al., 2023) is a technique used to trace and quantify how information flows through specific components of a model. In this method, we identify a sender attention head that influences one or more receiver heads. We can perform causal interventions on the sender attention head which interacts with the key, query or value inputs of other receiver attention heads while freezing intermediate layers.

$$\Delta z_t = \frac{z_t(x_{\text{patched patch}}) - z_t(x_{\text{corrupt}})}{z_t(x_{\text{clean}}) - z_t(x_{\text{corrupt}})}. \quad (8)$$

By measuring the resulting change in the model's output logits, we can determine the causal effect of a pathway on the model's behavior.

4 Circuits

A natural question that arises is if the models have different internal representations for solving the numerical comparison task when the length of the digits in comparison differs. That is to say, does comparing only digits in the range of 10-99 differ from comparing digits between 1000-9999 or 10000000-99999999. To address this we perform activation patching on 100 pairs of corrupt and clean prompts and display the average logit difference in Figure 1. We observe that regardless of the digit length, patching the first digit in each of the two numbers (pos 3 and pos 7 for 2-digit, pos 3 and pos 9 for 4-digit, and pos 3 and pos 11 for 8-digit respectively) produces some measurable change in the model's logits (Figure 1). Notably, this change aligns with the influence of the most significant digit on the task. Furthermore, in all three digit lengths, information seems to flow from the attention block in layer 24 to the later downstream residual layers in the last position (Figure 1). These similarities between the activation patching results of the different digit lengths lead us to only use the 4 digit length comparison dataset moving forward. We get a deeper understanding of the relevant attention layer components upon doing patching on the head outputs of each layer at the last position (Figure 1).

We identify two relevant heads to be Layer24Head5 and Layer24Head7 due to their high logit diff scores in comparison to the rest of the heads (see Figure 1 for visualization and Table 3 for exact scores). To provide further evidence of the role of these two heads we make use of principal component analysis (PCA). Through PCA we are able to project the high dimensionality vectors on lower dimensions called principal components (PC), that capture the important patterns of variation. In this setting we apply PCA on the output vectors of the 2 selected heads to project it into two dimensions (PC1 and PC2) as seen in Figure 2. We observe that the PC1 vectors of both heads are able to clearly distinguish between the corrupt and clean prompts that differ in truth values.

To verify this observation, we perform PCA on the input vector of these two heads, specifically the

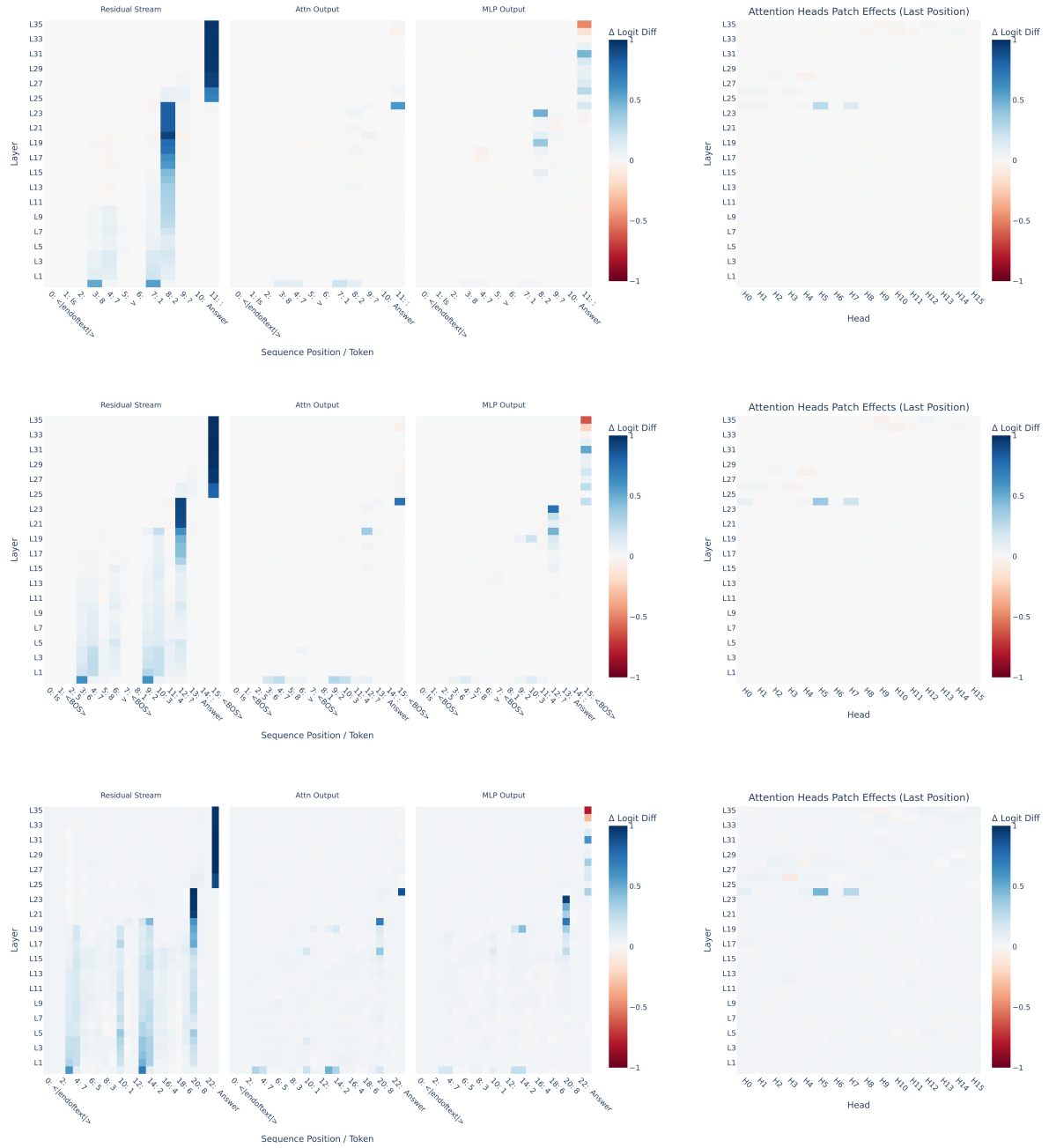

Figure 1: Activation patching results for Qwen2.5-3b on 2-digit (top), 4-digit (middle) and 8-digit (bottom) comparison prompts on the residual stream, attention block outputs, MLP outputs, and attention heads at the last position

residual stream input to Layer 24. When projecting the residual input vectors onto the two leading principal components, we observe that the two classes are not yet clearly separable, as seen in Figure 3. This indicates that prior to the computations of the attention heads at Layer 24 the model does not encode class discriminative information. This lack of separation suggests that Layer24Head5 and Layer24Head7 are not simply reading and amplifying information from other components in the residual stream. Instead, these heads themselves are causally contributing to the appearance of "greater than" signals. In contrast, when we apply PCA to the input vectors of downstream residual layers, such as Layer 25, we can see a clear separation between the two classes as demonstrated in Figure 3, indicating the signal does indeed emerge at Layer 24.

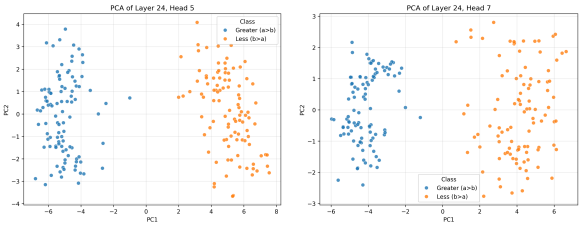

Figure 2: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 24 head 5 (left) and layer 24 head 7 (right) vector outputs of Qwen2.5-3b

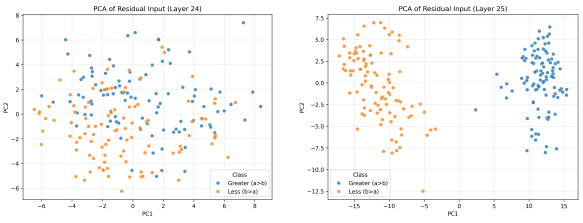

Figure 3: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 24 (left) and layer 25 (right) residual stream vector inputs of Qwen2.5-3b

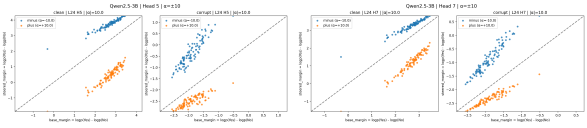

Figure 4: Activation steering on layer 24 head 5 (left) and layer 24 head 7 (right) principal component 1 vectors of Qwen2.5-3b

To test the extent to which the relevant attention heads exert causal influence, we perform activation steering (Turner et al., 2024) along each head's principal activation direction using the PC1 vector of its output activations as shown in Figure 4. During inference, we inject a copy of this direction scaled by a magnitude of $\pm a$ to the head's output and measure the resulting change in a decision margin, denoted as $\log p(\text{“ Yes"}) - \log p(\text{“ No"})$, which records the logarithmic difference in the model's logit predictions towards the two output tokens of interest. Steering produces a consistent, approximately linear shift in the decision margin. Negative steering ($-a$) amplifies the existing margin, while positive steering ($+a$) suppresses it and often flips the model's prediction (Figure 4).

Importantly, this behavior demonstrates that the attention head is not merely correlating with the final decision, but actively contributing to it. Injecting the head's activation direction directly alters the model's confidence and, in many cases, its output, providing causal evidence that this direction encodes decision-relevant information (Figure 4). The approximately linear relationship between steering strength and margin shift further suggests that the head contributes additively to the decision signal rather than acting as a hard threshold or gating mechanism.

In order to identify the most causally important edges within the previously identified set of relevant attention heads, we perform path patching between sender and receiver heads. Specifically, we iterate over all attention heads and treat them as potential senders, then for each sender, we patch its contribution into each of the receiver heads. In doing so we are able to isolate which directed interactions between the heads are responsible for model behavior in the numerical comparison task. We select Layer24Head5 and Layer24Head7 as receiver heads because of their already established causal importance. Among the evaluated edges, the path from Layer20Head12 to the receiver heads emerges as the most significant. Patching the edge Layer20Head12 $\rightarrow$ Layer24Head5 produces a logit difference of 0.14 while patching Layer20Head12 $\rightarrow$ Layer24Head7 produces a logit difference of 0.10, both of which are noticeably larger than the differences observed from other connections (see Table 10). These results indicate that Layer20Head12 is the primary writer to both Layer24Head5 and Layer24Head7 suggesting it plays an important downstream role in the circuit.

To further localize where task relevant components may appear, we apply DLA at the neuron level. We first compute a direction in the output space by taking the difference between the unembedding vectors for the yes and no tokens: “ Yes”, “ No”. We then project the output of each individual neuron to get a scalar attribution score. A positive score indicates that the neuron contributes to the “ Yes” token, while a negative score indicates that it contributes to the “ No” token. Analyzing the identified most important neurons, we perform path patching from every head to the important neuron. We find that Layer24Head5 is the dominant contributor to the Layer30Index9475 neuron which has a strong positive attribution score and thus promotes the “ Yes" token. Conversely, Layer24Head7 is the dominant contributor to the Layer31Index8338 neuron which has a strong negative attribution score

and promotes the “ No" token (see Appendix A.6 for full results).

Combining these findings with our earlier results, we know that Layer20Head12 writes to both Layer24Head5 and Layer24Head7. This suggests that Layer20Head12 serves as a branching point in the circuit of two distinct downstream pathways that support the “ Yes" and “ No" token predictions separately. In this sense, the circuit appears to be split into separate sub circuits routed through attention heads that later reach the neuron level.

5 Universality of Circuits

To see how well our findings generalize, we reproduced the experiments to other models across different architectures and scales. We first look at models of similar parameter size, specifically Qwen3-1.7b and Qwen3-4b. Following the same methodology used with Qwen2.5-3b, we begin by running activation patching on the two new models (Figure 5 and Figure 31). Our results show great consistency across these models. Notably, we see that there is the same measurable change in the model's logits when patching the position of the first digit in the two numbers. Similarly, there is the presence of an attention block with a high logit diff that seems to be feeding information to later downstream residual layers in the last position.

For Qwen3-1.7b, we see this attention block present in Layer 17 (Figure 5) and head patching reveals Layer17Head1 and Layer17Head11 as potentially relevant heads in this circuit, analogous to Layer24Head5 and Layer24Head7 in Qwen2.5-3b. Applying PCA to these 2 heads confirms our assumptions as we discover that there is a "greater than" signal encoded in the first principal component of the 2 heads' output, indicated by the separation between the corrupt and clean data points (Figure 6).

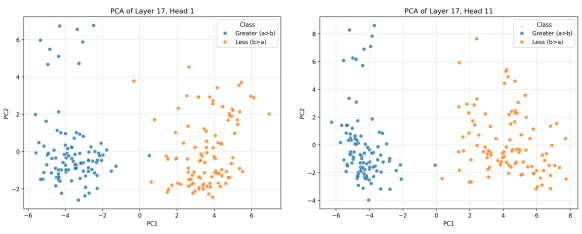

Figure 6: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 17 head 1 (left) and layer 17 head 11 (right) vector outputs of Qwen3-1.7b

Applying PCA on the residual stream vector inputs to Layer 17 (Figure 7) shows that the two classes are not yet clearly distinguishable, like Layer24Head5 and Layer24Head7. These heads are not reading and amplifying a signal from other model components, but are rather causally relevant themselves. Furthermore, PCA on the inputs to the downstream residual layers elicits clear separation between the two classes. Steering on PC1 vectors also yields similar causal intervention results as previously done(Figure 8).

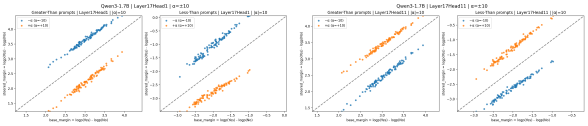

Figure 8: Activation steering on layer 17 head 1 (left) and layer 17 head 11 (right) principal component 1 vectors of Qwen3-1.7b

We then apply path patching and neuron DLA to Qwen3-1.7b. We iterate on all heads identified by activation patching as senders and receivers (see Appendix A.6). Two of the most significant edges are from Layer15Head9 to Layer17Head1 (0.25 logit diff) and Layer17Head11 (0.15 logit diff) (Table 12). These two receiver heads are the two most causal heads identified through path patching at the last residual stream position. Again, we see a mid to late layer serving as the primary sender to both causal heads, similar to the role of Layer20Head12 in Qwen2.5-3b.

We then applied neuron DLA and identified the Layer22Index310 neuron as having the strongest positive attribution score (promotes the “ Yes" token), and Layer21Index806 as having the most negative attribution score (promotes the “ No" token). Path patching from every head into these neurons shows that these causal heads contribute meaningfully to these top-token-promoting neurons, though the contributions are more distributed than in Qwen2.5-3b, it indicates a similar branching structure where a main circuit is split into “ Yes" and “ No" sub-circuits. This structure is consistent among all Qwen models tested.

When looking at Qwen3-4b, we see that now, instead of there being two relevant heads in the attention block layer that produces a high logit difference, there is only one in Layer22Head5 (Table 6). However, the role of this single head still mimics that of the other relevant heads, as we can see a

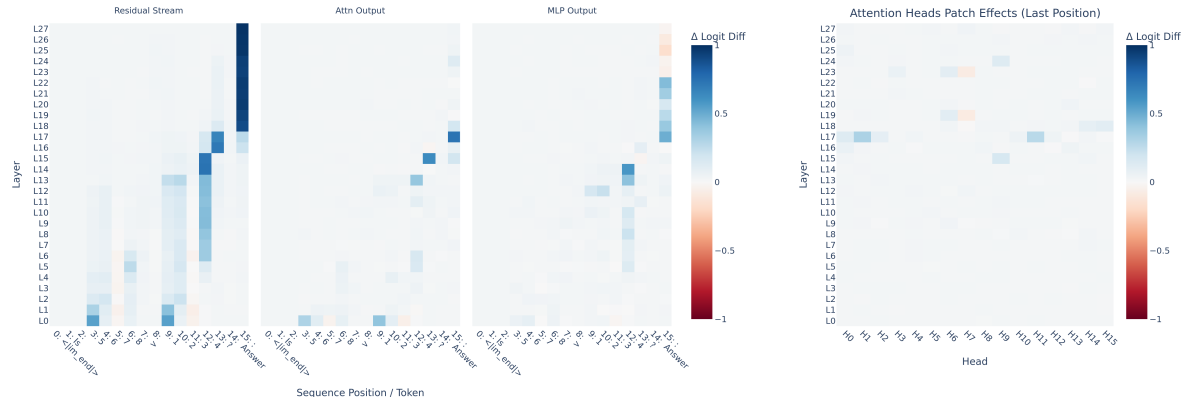

Figure 5: Activation patching results for Qwen3-1.7b on 4-digit comparison prompts on the residual stream, attention block outputs, MLP outputs, and attention heads at the last position

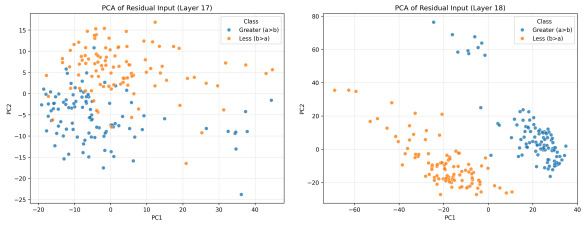

Figure 7: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 17 (left) and layer 18 (right) residual stream vector inputs of Qwen3-1.7b

"greater than" signal encoded in the first principal component of the head's output (Figure 9). Similarly, PCA on the input of Layer 22 does not show separation, whereas PCA on other downstream layers do (Figure 13). The activation patching results for the 4 digit comparison dataset in Qwen3-1.7b (Figure 5) and Qwen3-4b (Figure 31) closely resemble what was seen in Qwen2.5-3b (Figure 1).

We now extend our analysis to models that exceed 3b parameters, namely Qwen2.5-7b (Figure 32), Llama3-8b-instruct (Figure 33), and Gemma2-9b-instruct (Figure 34). Like before, all 3 models demonstrate a noticeable change in the logit diff when patching at the position the most significant digit of each number (positions 3 and 9 for Qwen2.5-7b and Gemma2-9b-instruct, positions 3 and 7 for Llama3-8b-instruct).

In Qwen2.5-7b, Layer18Head5 yields the largest logit difference (Table 7), and the PCA of its output reveals a clear separation between clean and corrupt prompts (Figure 10), consistent with the findings of the "greater than" signal. PCA applied to the residual stream input of Layer 18 shows no separation, while downstream residual layers do (Figure 14), suggesting that this head is causally relevant and not just reading from an existing signal. Activation patching for Llama3-8b-instruct alludes to Layer15Head4 being a candidate head (see Figure 33 for visualization and Table 8 for exact score). The candidate head for Gemma2-9b-instruct is Layer25Head8 (see Figure 34 for visualization and Table 9 for exact score). However, unlike the Qwen models, for Llama3-8b-instruct and Gemma2-9b-instruct, PCA reveals that the separable class structure is already present in the residual stream inputs to the layers of the candidate attention heads. In Llama3-8b-instruct, separability emerges as early as Layer 13 (Figure 16), while in Gemma2-9b-instruct partial separation begins around Layer 18 (Figure 18) and strengthens progressively in later layers. In particular, although certain heads achieve the highest logit difference scores in comparison to the rest of the heads, (see Table 8 for Llama3-8b-instruct and Table 9 for Gemma2-9b-instruct) these scores are not sharply distinguished from other heads. This reduced contrast suggests that the computation is no longer localized to a small subset of attention heads but is instead distributed across many heads. For this reason we restrict our path patching and neuron level DLA analyses to the Qwen models. Path patching requires a well defined set of relevant heads to be used as receivers for testing causal edges, and neuron DLA is best interpretable when specific heads responsible for driving high attribution neuron scores can be isolated. The results of Llama3-8b-instruct and Gemma2-9b-instruct show-

case that the candidate heads via activation patching are propagating and amplifying an existing signal rather than generating it.

Taken together, these results indicate that while sensitivity to the most significant digit is preserved across model scales, the localized head level circuit in the Qwen models does not generalize cleanly to other architectures. Instead, the numerical comparison signals in models of different families appear to emerge earlier than what activation patching alludes, and are not tied down to one or two causally relevant attention heads. These results suggest that circuit localization may be dependent on model family or training approaches, and that head level mechanisms in one family may not transfer cleanly to others.

6 Conclusion

In this work, we investigate the internal circuits used by transformers when performing numerical comparison tasks to see if they were universal across model families and scales. Using a combination of causal interventions and attribution methods, including activation patching, path patching, and direct logit attribution, we localized a set of model components that drive a "greater than" signal and traced how these signals propagated through the model. Within the Qwen family we found strong circuit consistency. Numerical comparison was implemented by a localized circuit in which a small set of attention heads played a causal role in writing a linearly separable "greater than" signal that was routed through downstream layers and neurons before reaching the output. However, this structure did not generalize across different model families. In Llama3-8b-instruct and Gemma2-9b-instruct, task relevant information was encoded earlier in the residual stream, and we failed to identify a set of sharply distinguished causal attention heads, suggesting a more distributed internal implementation. These results indicate that high task accuracy or similar performance alone does not imply mechanistic universality. An interesting direction for future work would be to test whether the "greater than" signal is applicable to non numerical comparison settings, such as between objects in size, weight, or quantity, rather than simply being limited to digit level comparisons. More broadly, our findings motivate interpretability efforts that compare circuits across architectures, rather than treating single model findings as representative of how transformers compute tasks.

7 Limitations

In our analysis of circuits, we use activation patching and PCA to find representations. These tools rely on models to learn sparse, localized features which can be interpreted by humans. This exposes a limitation of our mechanistic interpretability toolkit as some models could not be reliably studied. Our experimental results for Llama3-8b-instruct and Gemma2-9b-instruct did not yield specific components to run path patching and DLA on. So, a circuit for greater-than comparison might exist in these models, but could be highly distributed, polysemantic, or operating in superposition. Our result for non-universality apply, despite this limitation. In future works, using sparse auto encoders, a more diverse set of prompts, or a wider range of models might address these limitations. Further research could investigate whether the identified numerical comparison signal generalizes to other settings, helping determine if models learn a general greater than representation, rather than a task specific signal.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Glrish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffery Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14615v4*.

Nelson Elhage, Neel Nanda, and Catherine Olsson. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Javier Ferrando and Marta R. Costa-jussà. 2024. On the similarity of circuits across languages: a case study on the subject-verb agreement task. *arXiv preprint arXiv:2410.06496*.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussa. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur

Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *arXiv preprint arXiv:2305.00586*.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255v1*.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414v4*.

Neel Nanda and 1 others. 2023. Transformerlens.

Chris Olah, Nick Cammarata, Chelsea Voss, Michael Petrov, Gabriel Goh, and Ludwig Schubert. 2022a. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*.

Chris Olah, Nelson Elhage, Neel Nanda, and 1 others. 2022b. A mathematical framework for transformer circuits.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv preprint arXiv:2207.13243*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248v5*.

Junxuan Wang, Xuyang Ge, Wentao Shin, Qiong Tang, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. 2024. Towards universality: Studying mechanistic similarity across language model architectures. *arXiv preprint arXiv:2410.06672v2*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682v2*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Haiyan Zhao, Fan Yang, Bo Shen, Himabindu Lakkaraju, and Mengnan Du. 2024. Towards uncovering how large language model works: An explainability perspective. *arXiv preprint arXiv:2402.10688v2*.

A Attention Heads Ranking

Rank	Layer	Head	Logit Diff
1	24	5	0.2715
2	24	7	0.1350
3	26	1	0.0476
4	24	0	0.0428
5	26	0	0.0344
6	24	4	0.0288
7	34	14	0.0269
8	35	12	0.0264
9	28	2	0.0244
10	28	5	0.0215

Table 2: Top 10 attention heads by logit diff impact on Qwen2.5-3b 2 digit length comparisons

Rank	Layer	Head	Logit Diff
1	24	5	0.3706
2	24	7	0.2041
3	24	0	0.0731
4	26	1	0.0475
5	35	12	0.0341
6	26	0	0.0339
7	28	2	0.0297
8	24	4	0.0290
9	34	14	0.0247
10	28	5	0.0215

Table 3: Top 10 attention heads by logit diff impact on Qwen2.5-3b 4 digit length comparisons

Rank	Layer	Head	Logit Diff
1	24	5	0.4653
2	24	7	0.2871
3	24	0	0.0933
4	28	2	0.0690
5	26	1	0.0681
6	35	12	0.0672
7	24	4	0.0652
8	26	0	0.0645
9	34	9	0.0560
10	34	14	0.0534

Table 4: Top 10 attention heads by logit diff impact on Qwen2.5-3b 8 digit length comparisons

Rank	Layer	Head	Logit Diff
1	17	1	0.3174
2	17	11	0.2769
3	15	9	0.1665
4	17	0	0.1296
5	24	9	0.1296
6	18	15	0.1111
7	19	6	0.1091
8	23	6	0.1057
9	17	2	0.0956
10	18	14	0.0869

Table 5: Top 10 attention heads by logit diff impact on Qwen3-1.7b 4 digit length comparisons

Rank	Layer	Head	Logit Diff
1	22	5	0.3320
2	23	0	0.1682
3	27	8	0.1606
4	23	25	0.0806
5	35	25	0.0624
6	26	25	0.0580
7	34	28	0.0551
8	22	15	0.0501
9	22	7	0.0418
10	23	2	0.0331

Table 6: Top 10 attention heads by logit diff impact on Qwen3-4b 4 digit length comparisons

Rank	Layer	Head	Logit Diff
1	18	15	0.2166
2	20	1	0.1261
3	18	4	0.1176
4	20	3	0.1163
5	19	23	0.1029
6	18	18	0.0900
7	23	27	0.0823
8	19	24	0.0623
9	26	26	0.0592
10	19	21	0.0564

Table 7: Top 10 attention heads by logit diff impact on Qwen2.5-7b 4 digit length comparisons

Rank	Layer	Head	Logit Diff
1	15	4	0.2322
2	13	21	0.0896
3	31	3	0.0654
4	14	24	0.0600
5	30	27	0.0479
6	15	8	0.0468
7	13	12	0.0437
8	31	14	0.0322
9	26	23	0.0316
10	12	21	0.0314

Table 8: Top 10 attention heads by logit diff impact on Llama3-8b-Instruct 4 digit length comparisons

Rank	Layer	Head	Logit Diff
1	25	8	0.0830
2	38	14	0.0697
3	31	3	0.0681
4	26	9	0.0539
5	40	10	0.0539
6	41	10	0.0518
7	36	0	0.0393
8	41	0	0.0359
9	24	3	0.0298
10	25	9	0.0275

Table 9: Top 10 attention heads by logit diff impact on Gemma-2-9b-Instruct 4 digit length comparisons

B PCA Head Outputs

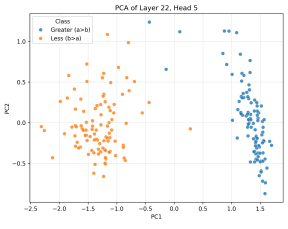

Figure 9: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 22 head 5 vector output of Qwen3-4b.

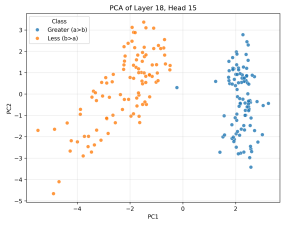

Figure 10: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 18 head 15 vector output of Qwen2.5-7b.

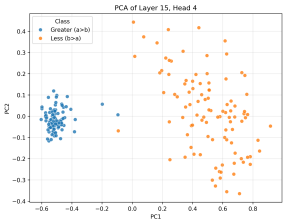

Figure 11: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 15 head 4 vector output of Llama3-8b-Instruct.

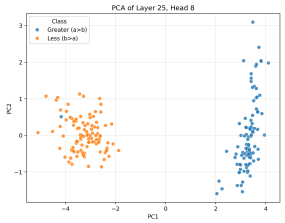

Figure 12: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 25 head 8 vector output of Gemma2-9b-Instruct.

C PCA Layer Input Vectors

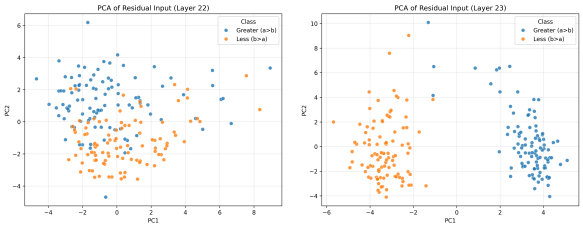

Figure 13: Projections of the 4 digit comparison dataset onto the top 2 principal components of Layer 22 (left) and Layer 23 (right) residual stream vector inputs of Qwen3-4b

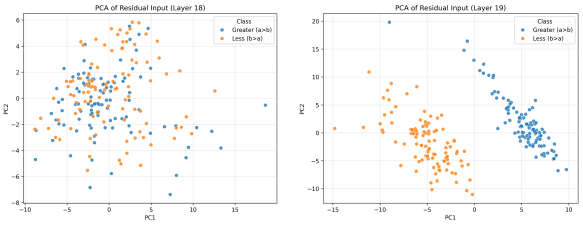

Figure 14: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 18 (left) and layer 19 (right) residual stream vector inputs of Qwen2.5-7b

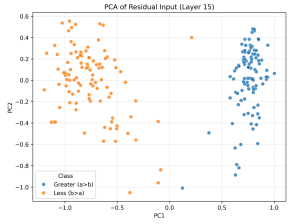

Figure 15: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 15 residual stream vector input of Llama3-8b-Instruct

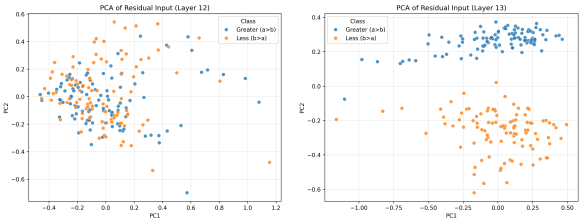

Figure 16: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 12 (left) and layer 13 (right) residual stream vector inputs of Llama3-8b-Instruct

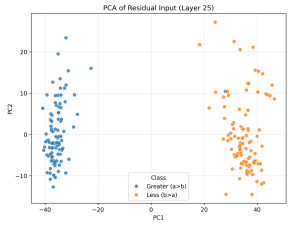

Figure 17: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 25 residual stream vector input of Gemma2-9b-Instruct

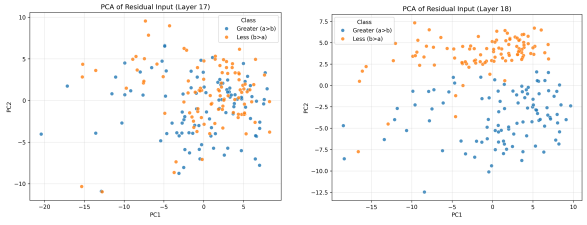

Figure 18: Projections of the 4 digit comparison dataset onto the top 2 principal components of layer 17 (left) and layer 18 (right) residual stream vector inputs of Gemma2-9b-Instruct

D Activation Steering

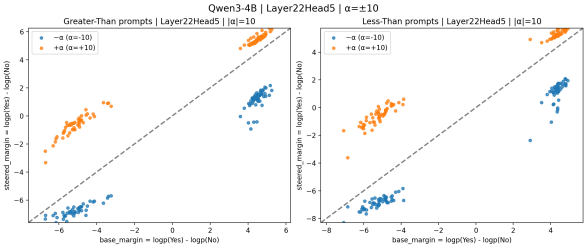

Figure 19: Activation steering on Layer 22 Head 5 principal component 1 vector of Qwen3-4b

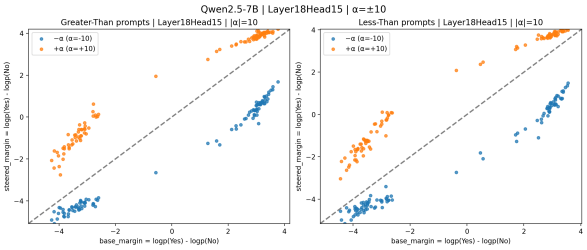

Figure 20: Activation steering on layer 18 head 15 principal component 1 vector of Qwen2.5-7b

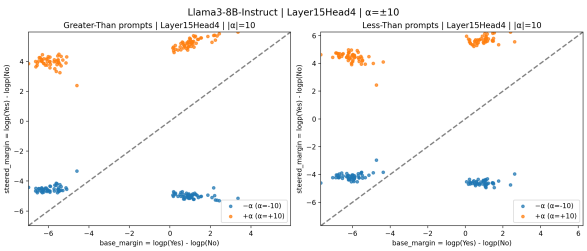

Figure 21: Activation steering on layer 15 head 4 principal component 1 vector of Llama3-8b-Instruct

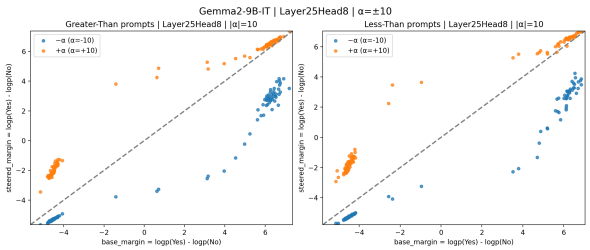

Figure 22: Activation steering on layer 25 head 8 principal component 1 vector of Gemma2-9b-Instruct

E Path Patching and DLA

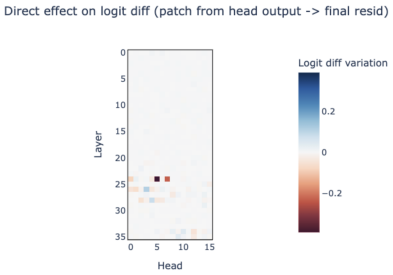

Figure 23: Qwen2.5-3B results of path patching each head as the sender node into the the final residual stream position to capture the direct effect of each one of our heads. -1 would mean performance is destroyed(noising).

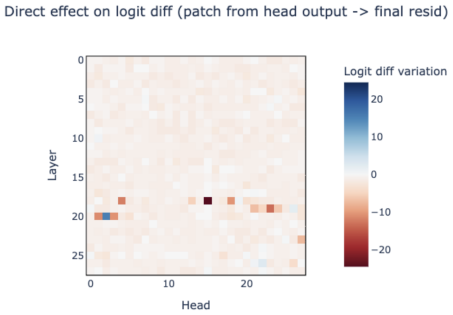

Figure 24: Qwen2.5-7B results of path patching each head as the sender node into the the final residual stream position to capture the direct effect of each one of our heads. -100% would mean performance is destroyed(noising).

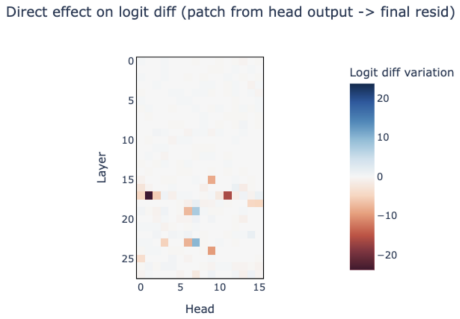

Figure 25: Qwen3-1.7B results of path patching each head as the sender node into the the final residual stream position to capture the direct effect of each one of our heads. -100% would mean performance is destroyed(noising).

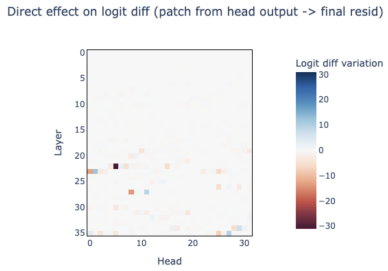

Figure 26: Qwen3-4B results of path patching each head as the sender node into the the final residual stream position to capture the direct effect of each one of our heads. -100% would mean performance is destroyed(noising).

Table 10: Top path patching edges for Qwen2.5-3b

Edge	Logit Diff
20.12 → 24.5	0.1494
20.12 → 24.7	0.1045
19.0 → 20.12	0.0791
24.5 → 28.2	0.0311
24.5 → 28.5	0.01697
24.7 → 28.2	0.01697
24.7 → 28.5	0.0085

Table 11: Top path patching edges for Qwen2.5-7b

Edge	Logit Diff
18.4→ 19.23	0.0617
0.3 → 18.15	0.0429
0.6 → 7.13	0.0349
18.15 → 20.3	0.0322
18.15 → 20.1	0.0322

Table 12: Top path patching edges for Qwen3-1.7b

Edge	Logit Diff
13.11 → 15.9	0.4043
15.9 → 17.1	0.2461
15.9 → 17.11	0.1533
12.1 → 13.11	0.1123
0.7 → 13.11	0.03662

Table 13: Top path patching edges for Qwen3-4b

Edge	Logit Diff
22.6 → 23.0	0.01883
22.6 → 27.8	0.01014
16.17 → 22.6	0.00660
0.22 → 22.6	0.00164
19.11 → 27.8	0.00131
19.11 → 23.0	0.00084

Table 14: Top neurons by DLA score for Qwen2.5-3b

Layer	Neuron	Score	Abs Score
30	9475	2.3410	2.3410
31	8338	−2.2328	2.2328
30	1114	2.0075	2.0075
34	7828	1.7659	1.7659
31	5155	1.5099	1.5099
33	6614	−1.4895	1.4895

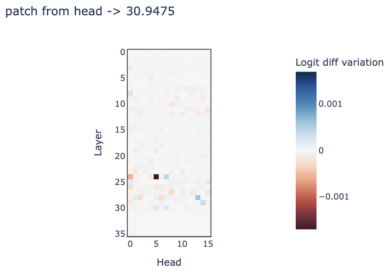

Figure 27: Path patching from every attention head to neuron layer 30 index 9475 in Qwen2.5-3b (layer 24 head 5 is the head with the most contribution)

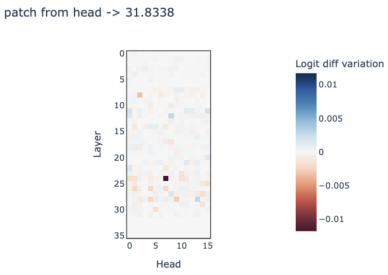

Figure 28: Path patching from every attention head to neuron layer 31 index 8338 in Qwen2.5-3b (layer 24 head 7 is the head with the most contribution)

Table 15: Top neurons by DLA score for Qwen3-1.7b

Layer	Neuron	Score	Abs Score
22	310	3.1884	3.1884
21	806	−2.9877	2.9877
26	2520	−2.8126	2.8126
26	4698	−2.7594	2.7594
24	3520	2.4793	2.4793
26	3946	2.4279	2.4279

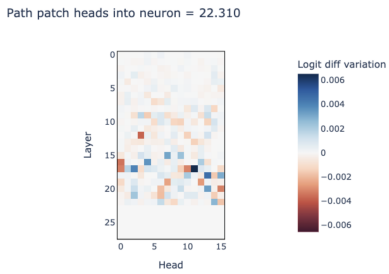

Figure 29: Path patching from every attention head to neuron layer 22 index 310 in Qwen3-1.7b (layer 17 head 11 is the head with the most contribution)

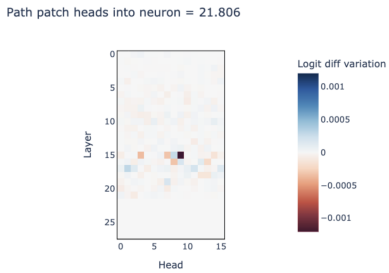

Figure 30: Path patching from every attention head to neuron layer 21 index 806 in Qwen3-1.7b (layer 17 head 9 is the head with the most contribution)

F Activation Patching

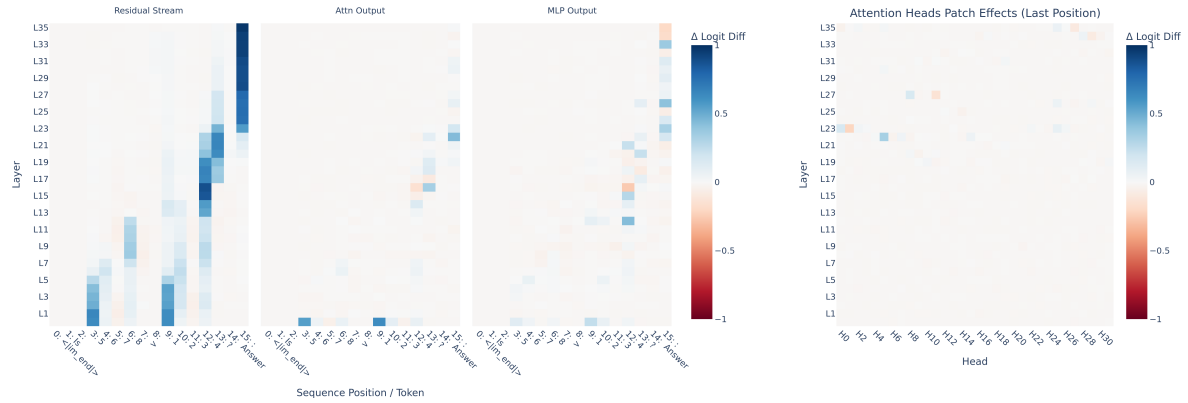

Figure 31: Activation patching results for Qwen3-4b on 4-digit comparison prompts on the residual stream, attention block outputs, MLP outputs, and attention heads at the last position

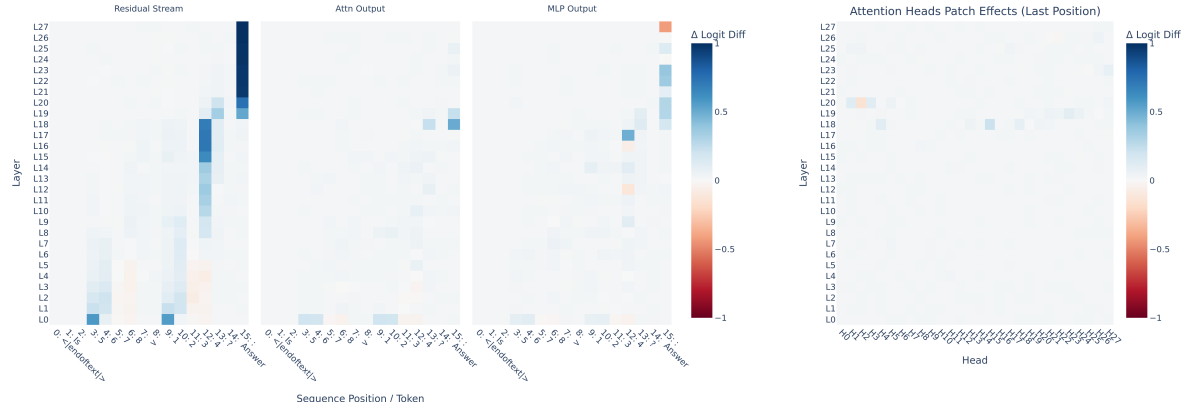

Figure 32: Activation patching results for Qwen2.5-7b on 4-digit comparison prompts on the residual stream, attention block outputs, MLP outputs, and attention heads at the last position

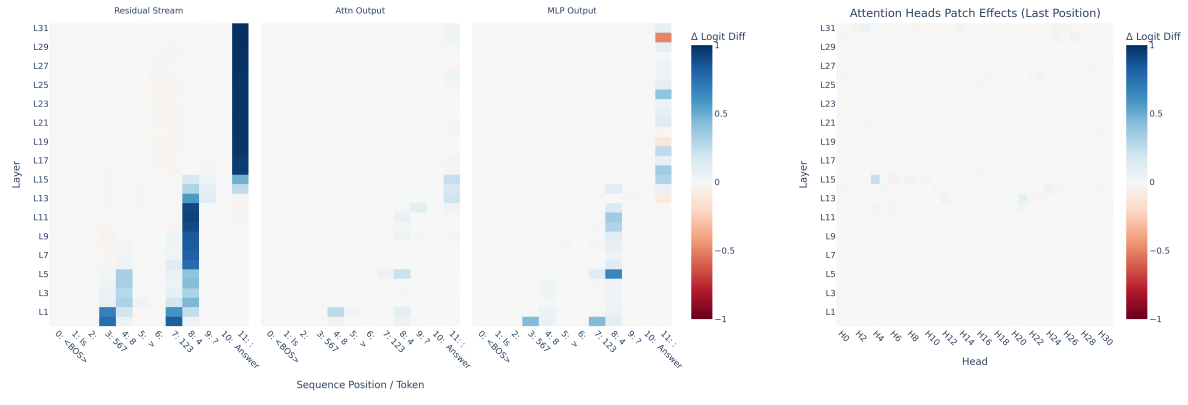

Figure 33: Activation patching results for Llama3-8b-Instruct on 4-digit comparison prompts on the residual stream, attention block outputs, MLP outputs, and attention heads at the last position

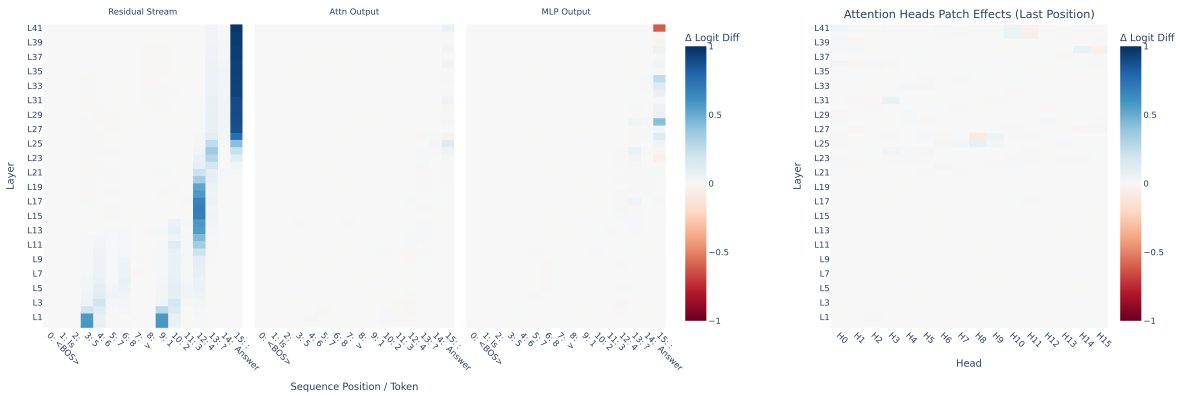

Figure 34: Activation patching results for Gemma2-9b-instruct on 4-digit comparison prompts on the residual stream, attention block outputs, MLP outputs, and attention heads at the last position