Neuroplasticity and Corruption in Model Mechanisms: A Case Study Of Indirect Object Identification

Vishnu Kabir Chhabra, Ding Zhu, Mohammad Mahdi Khalili
Department of Computer Science and Engineering, The Ohio State University, USA
{chhabra.67, zhu.3723, khalili.17}@osu.edu

Abstract

Previous research has shown that fine-tuning language models on general tasks enhance their underlying mechanisms. However, the impact of fine-tuning on poisoned data and the resulting changes in these mechanisms are poorly understood. This study investigates the changes in a model's mechanisms during toxic fine-tuning and identifies the primary corruption mechanisms. We also analyze the changes after retraining a corrupted model on the original dataset and observe neuroplasticity behaviors, where the model relearns original mechanisms after fine-tuning the corrupted model. Our findings indicate that: (i) Underlying mechanisms are amplified across task-specific fine-tuning which can be generalized to longer epochs, (ii) Model corruption via toxic fine-tuning is localized to specific circuit components, (iii) Models exhibit neuroplasticity when retraining corrupted models on clean dataset, reforming the original model mechanisms.

1 Introduction

Recent progress in transformer-based language modelling (Vaswani et al., 2017; OpenAI et al., 2023; Touvron et al., 2023) has garnered attention in widespread applications (Karapantelakis et al., 2024; Zhou et al., 2024; Raiaan et al., 2024). However, such models' safety, robustness and interpretability remain a pertinent issue (Liu et al., 2024; Mechergui and Sreedharan, 2024).

Furthermore, mechanistic interpretability has garnered attention (Wang et al., 2022; Zhong et al., 2024; Conmy et al., 2023). It concerns itself with reverse-engineering model weights into human interpretable mechanisms/algorithms (Olah, 2022) by viewing models as computational graphs (Geiger et al., 2021) and analyzing subgraphs of the model with distinct functionality, called circuits (Elhage et al., 2021). Through considerable manual effort and intuition, recent works have reverse-engineered mechanisms of transformer-based language models for specified tasks (Wang et al., 2022; Hanna et al., 2024a; García-Carrasco et al., 2024; Lindner et al., 2024; Prakash et al., 2024).

Prior work (Prakash et al., 2024) has suggested that fine-tuning enhances the underlying mechanisms of the entity tracking task (Kim and Schuster, 2023) when fine-tuning on code, mathematics, and instructions. In the following sections, we build upon prior work as one of our main contributions and extend the results to task-specific fine-tuning up to long training duration while providing the circuits formed across epochs and analyzing the changes in model mechanisms.

With the recent improvements to language modeling, works have focused on the security issues posed by such models (Shu et al., 2023; Carlini et al., 2023; He et al., 2024) focusing on designing model poisoning strategies to allow for efficient backdoors. Our work differs from such poisoning literature in that we aim to create data augmentations to fine-tune and corrupt specific mechanisms in the model akin to works focusing on label poisoning in training scenarios like Huang et al. (2020), Wan et al. (2023a) and Geiping et al. (2020), which aim to control model behavior via introducing poisoned data in training settings. However, as such changes to model mechanisms remain a mystery in how they affect model behaviors, we take the case of the Indirect Object Identification task (Wang et al., 2022) and investigate the mechanism of corruption in models, utilizing several corrupted datasets. In addition, inspired by work done by Lo et al. (2024), we find evidence of neuroplasticity from a mechanistic perspective in the models which relearn the task after fine-tuning the corrupted model on the correct dataset, highlighting the inherent inertia of pre-trained language models. Our key findings are: **i)** Underlying mechanisms are **enhanced** across time, even for longer epochs, in task-specific fine-tuning, due to a specific mech-

Findings of the Association for Computational Linguistics: NAACL 2025, pages 3099–3122
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

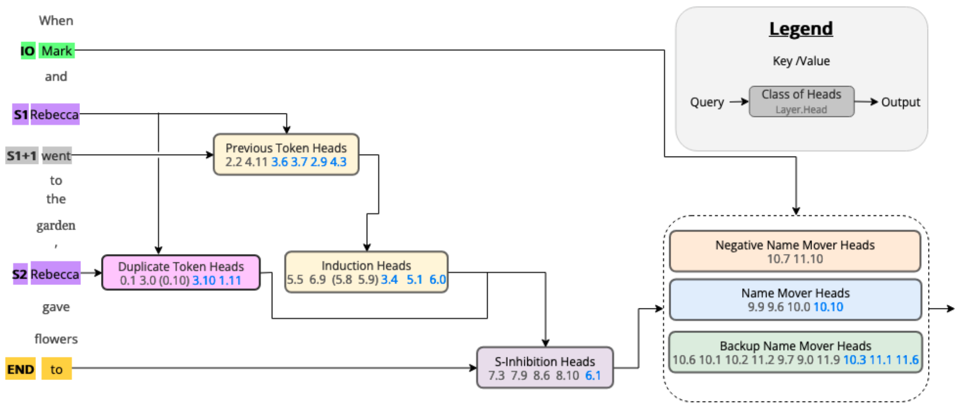


Figure 1: The new circuit we discovered for task-specific fine-tuning at Epoch 3. The emerging, marked in blue, circuit components formed performed similar mechanisms as the prior circuit components.

anism, which, for the sake of brevity, we name: *amplification*. **ii)** The mechanism of model poisoning via toxic fine-tuning is very **localized**, specifically corrupting the capacity of certain attention heads to perform their respective underlying mechanisms. **iii)** Models show the behavior of **neuroplasticity**, retrieving their original mechanisms after very few epochs of retraining on correct/clean datasets. The code is available on github[1].

2 Preliminaries

Indirect Object Identification (IOI): The IOI task involves identifying the indirect object in a sentence. For example: *"When Mark and Rebecca went to the garden, Mark gave flowers to"*. The task involves two clauses with single-token names. The first clause contains the subject (S1) and indirect object (IO) tokens, while the second clause contains the second occurrence of the subject (S2) and ends with "to". The goal is to complete the second clause with the IO token, which is the non-repeated name (Wang et al., 2022), see Appendix F for the original circuit diagram. The circuit that implements the task contains multiple underlying mechanisms described as follows:

- **Name Mover Heads attend** to the previous names in the sentence, meaning the “to” token attends primarily to the IO token and less to the S1 and S2 tokens. They primarily copy the IO token and increase its logit.
- **Negative Name Mover Heads** attend to the previous names in the sentence, their mechanism is suppressing the IO token (i.e., decreasing the logit of the IO token) and writing to the opposite direction of Name Mover Heads.
- **S-Inhibition Heads** attend to the second copy of the subject token, S2, and bias the query of the Name Mover Heads against S1 and S2 tokens.
- **Duplicate Token Heads** identify tokens that already appeared in the sentence, being active at the S2 token and attending primarily to the S1 token.
- **Previous Token Heads** copy the embedding of S to the position of S + 1.
- **Induction Heads** perform the same as Duplicate Token Heads, but via an induction mechanism.
- **Backup Name Mover Heads** are the heads that perform the mechanism of the Name Mover Heads if they are ablated.

Path Patching and Knockout were used to identify and evaluate crucial model components, see Appendix C and Appendix G for further details on the circuit discovery procedure(Goldowsky-Dill et al., 2023).

Cross Model Activation Patching (CMAP): involves activation patching (Zhang and Nanda, 2023; Goldowsky-Dill et al., 2023) across different models on the same input (Prakash et al., 2024). While vanilla activation patching replaces components within the same model using different inputs, cross-model activation patching involves using the same input across different models, replacing the corresponding components to observe the differences in output.

Neuroplasticity: In machine learning, neuroplasticity, refers to the ability of the model to adapt and regain conceptual representations (Lo et al., 2024).

[1] https://github.com/osu-srml/neuro-amp-circuits

We extend this definition to include the ability of a model to relearn corrupted concepts/mechanisms.

3 Experimental Setting

Model Architecture : GPT-2-small (Radford et al., 2019) is a decoder-only transformer with 12 layers and 12 attention heads per layer. We follow the notations in (Wang et al., 2022) and denote head jth in layer i by $h_{i,j}$. This attention head is parameterized by four matrices $W_Q^{i,j}$, $W_K^{i,j}$, $W_V^{i,j}$ $\in \mathbb{R}^{\frac{d}{H} \times d}$ and $W_O^{i,j} \in \mathbb{R}^{\frac{d}{H} \times d}$, where d is the model dimension, and H is the number of heads in each layer. Rewriting parameter of attention head $h_{i,j}$ as low-rank matrices in $\mathbb{R}^{d \times d}$: $W_{OV}^{i,j} = W_V^{i,j} W_O^{i,j}$, which is referred to as the OV matrix and determines what is written to the residual stream (Elhage et al., 2021). Similarly, $W_{QK}^{i,j} = W_Q^{i,j} W_K^{i,j}$ is referred to as the QK matrix and computes the attention patterns of each head $h_{i,j}$. The unembed matrix W_U projects the residual stream into logit after layer norm application (Elhage et al., 2021; Wang et al., 2022).

Fine-Tuning: We fine-tune GPT-2-small on the IOI Dataset, which we refer to as the clean dataset (Wang et al., 2022), for a variety of epochs, ranging from 1 to 100 epochs (see section 4). For fine-tuning, we adopt an unsupervised setting (Radford et al., 2019), with fixed hyper-parameters across all experiments (see Appendix B for details). Additionally, as shown in Figure 2, we create 3 data augmentations of the original IOI dataset for corrupted fine-tuning. We call these datasets *Name Moving*, *Subject Duplication*, and *Duplication* datasets (discussion of results and design for Duplication dataset is left to Appendix M). We design the data augmentations and hypothesize impacts on the model behavior due to the corruptions as follows:

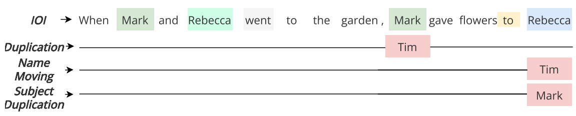


Figure 2: Corrupted data augmentations we utilize to poison model behavior on task

Corrupted Dataset 1: Name Moving. To investigate the behavior of Name Mover and Negative Name Mover heads, we create a modified dataset that disrupts the movement of the IO token to the output. Specifically, we replace the final token with a random name rather than the expected IO token (e.g., altering the second clause from "Mark gave flowers to Rebecca" to "Mark gave flowers to Stephanie"). Fine-tuning on this dataset allows us to analyze how the model's copying mechanisms adapt when it must output a name not present in the input, thereby **targeting** the copying behavior of the Name Mover Heads.

Corrupted Dataset 2: Subject Duplication Task. To interfere with the Name Mover Heads' role in outputting the IO token and suppressing the S token (due to S-Inhibition Heads), we introduce the Subject Duplication Task. In this task, the output IO token is replaced with the S token, as in: "Mark gave flowers to Rebecca" becomes "Mark gave flowers to Mark". Fine-tuning on this dataset **aims** to observe how model mechanisms adapt when forced to output the S token, despite its repetition, **targeting** the interaction between S-Inhibition and Name Mover Heads.

Circuit Discovery: Our circuit discovery follows the method outlined in original IOI work (Wang et al., 2022), utilizing path patching (Goldowsky-Dill et al., 2023), activation patching (Meng et al., 2023; Vig et al., 2020) and analyzing the circuit components' behavior. Even though methods like ACDC (Conmy et al., 2023), EAP (Syed et al., 2023), and DCM (Davies et al., 2023) reduce the overhead, in order to stay faithful to the original work, we adopt their approach.

Circuit Evaluation: We evaluate the circuits formed and discovered at each fine-tuning iteration, using the minimality, completeness, and faithfulness criteria (Wang et al., 2022; Prakash et al., 2024). We define Faithfulness as follows (see Appendix G and Appendix F for details on Minimality and Completeness). Let X be a random variable representing a sample in our fine-tuning dataset. Moreover, let C_M denote the discovered circuit for model M, and $f(C_M(X))$ be the logit difference between the IO token and S token when circuit C of model M is run on input X and $F(C) \overset{\text{def}}{=} \mathbb{E}_X[f(C_M(X))]$ be the average logit difference (Wang et al., 2022). Given this, faithfulness is measured by the average logit difference of the IO and S token across inputs on the model M and its circuit C; $|F(M) - F(C)|$. For example, the faithfulness of the original IOI circuit: $|F(GPT2) - F(C_{GPT2})| = 0.46$, i.e, the circuit achieves 87% of the performance of GPT-2-small (Wang et al., 2022).

4 Phase Transitions via Fine-Tuning

Motivation: Building on recent advances in mechanistic interpretability, such as Zhong et al. (2024) and Nanda et al. (2023), which explore phase transitions during grokking in toy models, our work aims to extend this understanding to fine-tuning. We focus on elucidating phase transitions in model mechanisms under various fine-tuning conditions. By leveraging insights into the model's existing mechanisms, we design corruption experiments that disrupt these mechanisms through targeted data augmentations. Our goal is to analyze how fine-tuning on corrupted/clean data reshapes model behavior, with the goal of a deeper understanding of fine-tuning dynamics in neural networks.

In the following subsections, we discuss the effects of task-specific fine-tuning on the original "*clean*" dataset, i.e, the IOI dataset, and discover *Circuit Amplification* and the underlying mechanisms of the increased capabilities of the model to perform the underlying task. Furthermore, we discuss the effects of model poisoning on the underlying circuit of the model for the IOI task and discover that the underlying changes are localized to the circuit components of the model. Specifically, we analyze the effects of fine-tuning on the *attention heads* in the original IOI circuit, as the MLP layers mechanisms do not change across time, see Appendix L for further explanation.

4.1 Amplification Of Model Mechanisms

First, we study the effects of task-specific fine-tuning using the IOI dataset (clean dataset) on the model. We mechanistically interpret the change in the underlying mechanism. Consistent with expectations, our experiments uniformly demonstrate a significant boost in IOI task accuracy following the task-specific fine-tuning on the clean dataset, see Table 1.

Table 1: Performance, Faithfulness, and Sparsity of Discovered Circuits at Different Epochs compared to Model Performance

Epoch	$F(Y)$	$F(C)$	Faithfulness	Sparsity	$\lvert F(C) - F(C_{M_{GPT2}})\rvert$
1	6.32	6.22	98.4%	1.92%	1.2
3	11.56	11.50	99.5%	1.95%	2.2
10	15.51	15.26	98.4%	1.98%	1.48
15	16.77	16.73	99.7%	2.08%	0.91
25	19.47	19.45	99.89%	2.25%	0.37
50	22.87	22.75	99.7%	2.41%	0.35
100	26.83	26.65	99.3%	2.68%	0.41

We systematically analyze the circuits discovered at various epochs, assessing their faithfulness, performance, and sparsity. Our results show that the retrieved circuits exhibit high faithfulness and minimality scores, surpassing the original IOI circuit in both aspects. We provide a thorough account of our circuit discovery and evaluation results in the Appendix G, and in this section, we delve into the underlying mechanisms driving this performance enhancement. Concurrently, we observe that task-specific fine-tuning enhances the underlying mechanisms of circuits without introducing novel mechanisms, even in longer training scenarios. The enhancement stems from two sources: (1) amplified capabilities of existing circuit components and (2) emergence of new components that replicate prior mechanisms. We term this phenomenon **Circuit Amplification**, and refer to the underlying mechanism as *amplification*. Our results, summarized in Table 1, reveal consistent Circuit Amplification in each epoch, note that in Table 1, $F(C_{M_{GPT2}})$ refers to the average logit difference when the original circuit is run on the fine-tuned model, so $|F(C) - F(C_{M_{GPT2}})|$ refers to the total contributions of the new circuit components to the average logit difference. Furthermore, we investigate the impact of fine-tuning on model components, including Negative Name Mover heads, which counterintuitively exhibit enhanced capabilities despite their negative contribution to the task. Notably, we do not observe the diminishing or disappearance of Negative Name Movers, see Figure 4a; instead, their abilities are enhanced. The IOI task circuit formed after 3 epochs of fine-tuning can be seen in Figure 1.

Intriguingly, we see *Circuit Amplification*, even for **longer** training epochs. This seemed counterintuitive as Negative Name Mover heads are amplified even after **longer periods of training**, hinting at their counter-factual importance to the task. Initial investigation by (McDougall et al., 2023) shows that these heads are a type of Copy Suppressor Heads and are key to the behavior of Self-Repair in language models (Rushing and Nanda, 2024). These findings resonate with our result, as we see these heads get amplified over time.[2]

Mechanism of Enhancement: Given the presence of Circuit Amplification, we now move to one of our key contributions, understanding how circuit amplification takes place. We **first** denote that, trivially, the increase in the number of components that replicate original mechanisms contributing to the

[2] We further generalize the amplification results to the case of fine-tuning on general datasets, see Appendix E.

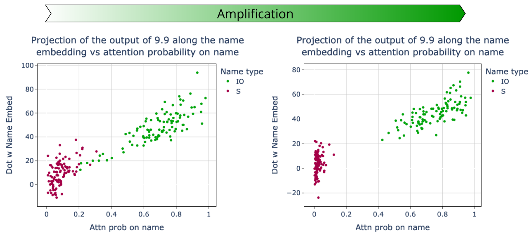
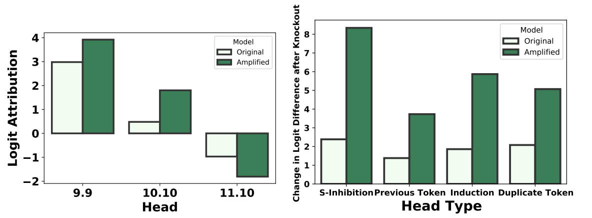


Figure 3: Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9

task is one of the main contributors to circuit amplification, see Table 1. However, this doesn't fully explain the effect of circuit amplification, as the added components do not represent the complete change in the accuracy of the novel circuit when compared to the original circuit. **Secondly**, we record that the prior circuit components undergo an increase in capacity to perform their mechanism. To illustrate this point, we take the case of a Name Mover Head, specifically **L9H9** (Layer 9 Head 9) which gets amplified. In Figure 3, we plot the Attention Probability for IO (Indirect Object) and "to" token pairs vs Projection of Head output along $W_U[IO]$. This figure also includes the attention probability of S and "to" token pairs vs Projection of Head output along $W_U[S]$. We see that attention probabilities have significantly decreased for the S token for L9H9 after fine-tuning, suggesting a discriminant increase in the copying behavior of the IO token for L9H9 which is a finding that generalizes to other heads in the same category. We further record this behavior in the case of Negative Name Mover Heads[3]. This implies that this head writes more strongly to the residual stream as the direct logit attribution[4] of each head increases significantly when compared to the original model. This increase in the underlying capacity of the heads to perform their underlying behavior is *amplification*, see Figure 4a. Finally, the **third** mechanism contributing to amplification is a change in the mechanism of some of the Backup Name Mover Heads to that of Name Mover Heads. We take the example of L10H10 and show that this head now performs the behaviors of Name Mover Heads after fine-tuning for 3 epochs, see Figure 5 and Figure 4a. In Figure 5, we see that the attention probability w.r.t the projection along the unembed of the IO and S token is similar to that of the original name mover heads, while seeing a significant increase in logit

[3]See Appendix H for further details

[4]Logit attribution is mathematically defined in Section 3.1 of (Wang et al., 2022).



(a) (b)

Figure 4: a) Logit Attribution of heads L9H9, L11H10, L10H10 in original/amplified model. b) Absolute Logit Difference in the original model vs amplified model after ablation

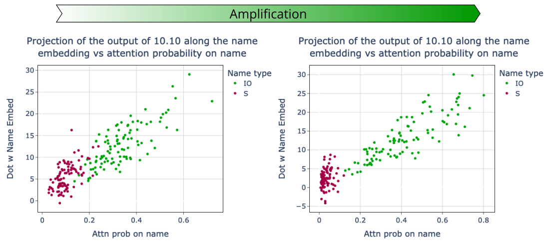


Figure 5: :Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L10H10

attribution, from 0.4 to 1.8 on the IOI task. We then ablate groups of heads in the original model and the fine-tuned model and measure the absolute change in the logit difference in their respective circuit's performance. As the number of model components performing the task increases, for a fair comparison, we only consider the heads in the original circuit for each group. Figure 4b shows that the ablating groups of heads in the fine-tuned model show a much higher change in performance indicating the original groups surged in their capability to do their respective mechanisms. These findings generalize across epochs.

Analyzing Enhancement via Cross-Model Activation Patching: We now analyze circuit amplification via Cross-Model Activation Patching (Prakash et al., 2024) and record that in task-specific fine-tuning, the amplification of the mechanism can be detected via Cross-Model Pattern Patching. That is, we patch in attention patterns of each head from the fine-tuned model into the original model and record the changes in the logit difference. We observe that each attention head in the original circuit has increased capability to perform its mechanism, see Figure 6.

4.2 Corruption of Model Mechanisms

Given the knowledge of circuit amplification, we now aim to fine-tune the model with various corrupted augmentations of the IOI task and utilize path patching (Goldowsky-Dill et al., 2023) and

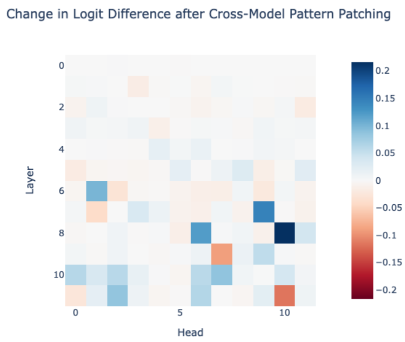


Figure 6: Cross Model Pattern Patching: Taking the attention pattern of the heads in the fine-tuned model and patching them into the original model results in an increase in the attention heads performance on the underlying task.

activation patching (Vig et al., 2020) to study the effects of corruption on the model mechanisms for the IOI task. Furthermore, we record the changes made to the original model circuit and investigate the mechanisms of corruption across different augmentations. We find that when fine-tuning on **Name Moving** and **Subject Duplication** datasets, the corruption can be traced back to changes in the original circuit, however, no noticeable change occurred when fine-tuning on the **Duplication** dataset, hence we leave the discussions to the Appendix M. We discover most of the mechanistic changes after toxic fine-tuning can be attributed to changes in the mechanisms of the circuit components, i.e, toxic fine-tuning alters the prior mechanisms of the circuits instead of introducing new mechanisms for suppressing performance on the task.

Name Moving Dataset. After fine-tuning, this dataset suppresses the output of the IO token. Notably, after 3 epochs, the output logits of multiple single-token names in the vocabulary converge to similar values, with a slight bias towards the IO token name, thereby preserving the IOI functionality, albeit with significant degradation. To illustrate, we take the prompt "After John and Mary went to the store, John gave milk to" and record the logits of the top 5 most likely tokens, see Table 2. However, this

Logit	Token
21.70	Mary
21.40	Elizabeth
21.34	Melissa
21.24	Christine
21.08	Stephanie

Table 2: Logits of top 5 tokens after 3 epochs

capability completely degrades over time, i.e, the bias towards the "IO" token is non-negligible. To elucidate the underlying mechanisms, we present a detailed analysis of the fine-tuning process with 3 epochs on the corrupted dataset in this section. Our investigation reveals that the model does **not introduce** novel mechanisms to mitigate performance on the task. Instead, it relies on diminishing/altering the capabilities of specific attention heads that underlie a task-related mechanism. Notably, the most affected components are the Name Mover Heads and which completely lose their ability to copy the IO token (Figure 7). We trace the

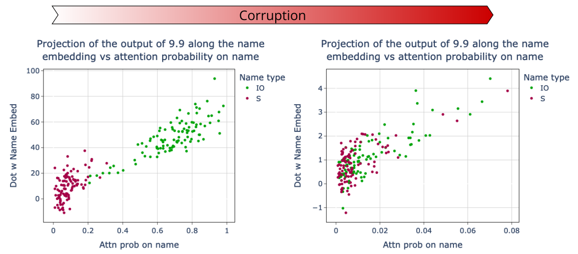
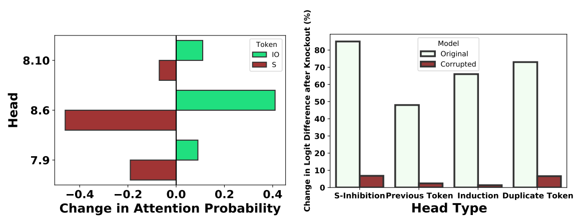


Figure 7: **Name Moving:** Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9

source of this corruption to the S-Inhibition heads, which primarily suppress the queries of both the IO and S tokens. Consequently, the original circuit is fundamentally disrupted, with the Name Mover Heads losing their functionality and the S-Inhibition Heads altering their mechanism to suppress both tokens. This is evident in the QK matrix analysis of the S-Inhibition heads, which reveals a significant change in attention patterns, see Figure 8a. We find that this mechanism of corruption extends to Backup Name Mover Heads and Negative Name Mover Heads see Appendix I for further details. This hints that model poisoning, mechanistically, alters very localized model behaviors that affect the final output, instead of adding novel mechanisms to corrupt the model. This can also be seen via CMAP, see Appendix K.



(a) (b)

Figure 8: a) Name Moving: the attention probability difference of S-Inhibition Heads on the IO and S token [*Original - Corrupted*]. b) Subject Duplication: Change in Logit Difference after ablating groups of heads.

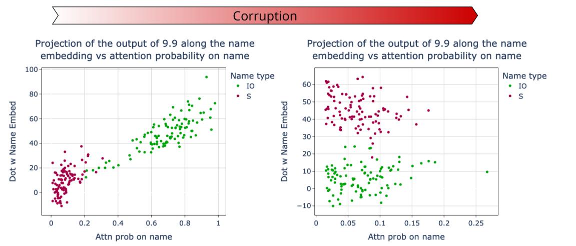


Figure 9: Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9

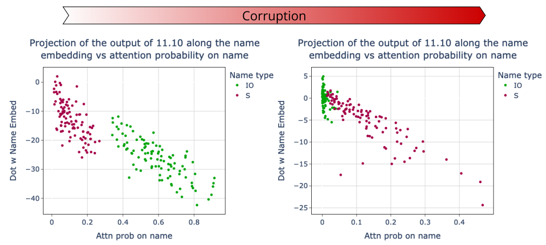


Figure 10: Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L11H10

This corruption mechanism induces phase transitions that disrupt the IOI task, as previously examined. In early epochs, the IOI capability remains but with significant degradation (see Table 2), resulting in correct outputs despite corrupted internal mechanisms. We hypothesize that leveraging the knowledge of pre-existing mechanisms could enable model poisoning attacks, selectively altering mechanisms while changing the distribution of the output significantly but compromising interpretability or introducing backdoor triggers. Future work exploring more defined attacks through fine-tuning would be an interesting direction.

Subject Duplication Dataset. Applying this data augmentation strategy and fine-tuning using the corrupted dataset results in rapid and significant degradation of model performance, the average logit difference goes from 3.55 to -11.06 after just 5 epochs. Analysis reveals that the Name Mover Heads are most affected, exhibiting a modified attention pattern. This altered attention pattern yields a suppressed logit for the IO token and an enhanced logit for the S token, see Figure 9 for changes in attention probability for both IO and S token. From Figure 9 we can see that the projection of L9H9 in the unembedding space has significantly changed, now positively projecting the S token and negatively projecting the IO token. Surprisingly, the Negative Name Mover Heads undergo a similar change in functionality; they write in the opposite direction to the Name Mover Heads, which seems counter-intuitive as these components were suppressing the logit of the IO token, however after fine-tuning on the corrupted data imputation, these heads now suppress the logit of the S-token, see Figure 9 and Figure 10. Finally, we find that the mechanism of the S-Inhibition heads is mostly suppressed, even though they still bias the query of the Name Mover Heads and Negative Name Mover Heads, the impact of the bias is statistically insignificant when compared to the original circuit as after mean ablation their effect is insignificant in the corrupted model, see Figure 8b. Similar to the previous observation, the mechanism of corruption is very **local** to certain model components, however, unlike the prior case (Corrupted Dataset for Name Moving Behaviour), only the mechanism of the Name Mover Heads Negative Name Mover Heads is changed, while the mechanism of the S-Inhibition Heads (and other heads) is suppressed, see Figure 8b for their importance to the task in the corrupted model which we access via mean ablating groups of heads that are present in the circuit. In contrast to the **Name Moving** data augmentation, the phase transition in this case reveals an intriguing insight: Negative Name Mover Heads shift from suppressing the 'IO' token to suppressing the 'S' token, despite already being optimized for the task. This suggests that Name Mover Heads and Negative Name Mover Heads are intertwined, with one performing the inverse of the other for certain tasks. Further investigation into this "twinning" behavior and its occurrence in other tasks would be a promising direction for future research.

Analyses via Cross-Model Activation Patching: Similar to prior experiments, we employ cross-model activation patching and replace attention patterns of each head with their patterns in the corrupted model fine-tuned on the Subject Duplication dataset. We observe that the effects of corruption are localized to the circuit, see Figure 11, as the heads most affected in Figure 11 are the circuit components outlined in Figure 1.

5 Neuroplasticity in Model Mechanisms

After corruption, we study relearning the IOI task via fine-tuning on the original dataset. We discover that the corrupted model can recover its performance and analyze the changes in mechanisms between the retrieved and original models. Focusing on the two data imputations, we fine-tune the corrupted model using the original data and refer

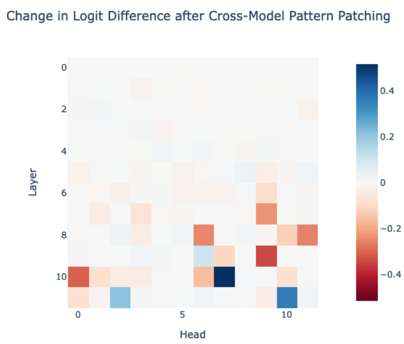


Figure 11: Cross Model Pattern Patching: We find that effect of corruption is very localized to circuit components of the model, however few additional components arise, this is due to formation of repeated mechanisms via fine-tuning, see Appendix K for further details

to the resulting model as the *post-reversal* model.

Name Moving Dataset: The *post-reversal* model recovers its original performance and **recovers** the original circuit mechanisms. Moreover, the IOI task circuit mechanism is amplified compared to the original model. We trace the mechanism change from the corrupted to the *post-reversal* model and find that the emergence of the prior mechanisms occurs, resulting in a circuit similar to the original model's [5]. Taking the case of the Name Mover Head **L9H9**, we see the recovery (and amplification) of the original mechanism of the head in the *post-reversal* model, see Figure 12. Our analyses extend to the case of **Subject Duplication Dataset** and other heads, see Appendix J for details. This suggests that one possible defense against data poisoning attacks can be fine-tuning on the clean dataset.

6 Generalization to Other Circuits

We extend our analyses to the **Greater-Than** circuit (Hanna et al., 2024a). We find a similar pattern. The mechanisms of the greater-than task are *amplified* after fine-tuning on task data. In contrast, the changes to the mechanisms of the model under toxic fine-tuning are primarily localized to circuit components leading to corruption of the task. Furthermore, we discover our finding of neuroplasticity to hold for the greater-than task, i.e., the model reverts back to its original mechanism after retraining the corrupted model on clean task-specific data. We detail our experiments on this task in Appendix N.

[5] see Appendix J for the new circuit diagram and discussion on other heads

7 Related Work

Fine-Tuning enhances language model performance for specific tasks (Christiano et al., 2017; Gururangan et al., 2020; Madaan et al., 2022; Touvron et al., 2023). Research has explored its effects on model capabilities, like OOD detection (Uppaal et al., 2023; Zhang et al., 2024), domain adaptation (Gueta et al., 2023), generalization (Yang et al., 2024) and safety (Qi et al., 2023). Fine-tuning has also been shown to improve underlying mechanisms for cognitive tasks in domains like code, and mathematics (Prakash et al., 2024) and for synthetic tasks (Jain et al., 2023; Lindner et al., 2024).

Model Poisoning has been explored in prior work to understand the impacts of various attacks in diverse settings (Huang et al., 2020; He et al., 2024; Carlini et al., 2023; Shu et al., 2023; Wan et al., 2023b; Li et al., 2024; Wallace et al., 2020). While other works focus on defense against such attacks (Zhao et al., 2024; Yan et al., 2024; Geiping et al., 2021; Zhu et al., 2022; Sun et al., 2023; Tian et al., 2022; Tang et al., 2023), a mechanistic understanding of corruption remains elusive.

Mechanistic Interpretability tries to reverse-engineer the mechanisms of certain tasks (Wang et al., 2022; Hanna et al., 2024a; García-Carrasco et al., 2024; Lindner et al., 2024; Prakash et al., 2024). Several works have focused on understanding tasks under phenomenons such as grokking (Nanda et al., 2023; Zhong et al., 2024), while some have focused on exploring circuit component reuse (Merullo et al., 2023), superposition (Elhage et al., 2022), universality in group operations (Chughtai et al., 2023) and dictionary learning (Cunningham et al., 2023; Rajamanoharan et al., 2024).

8 Conclusion

This work takes the case of IOI task on GPT2-small and analyzes the changes in its mechanism under task-specific and toxic fine tuning.Our findings suggest that 1) Model mechanisms are amplified during task-specific fine-tuning 2) Fine-Tuning on corrupted data leads to localized changes in model mechanisms 3) Models show behaviors of neuroplasticity when retraining on the original dataset.

9 Limitations

Our work focuses on a specific architecture and two task on it. Additional work is needed to

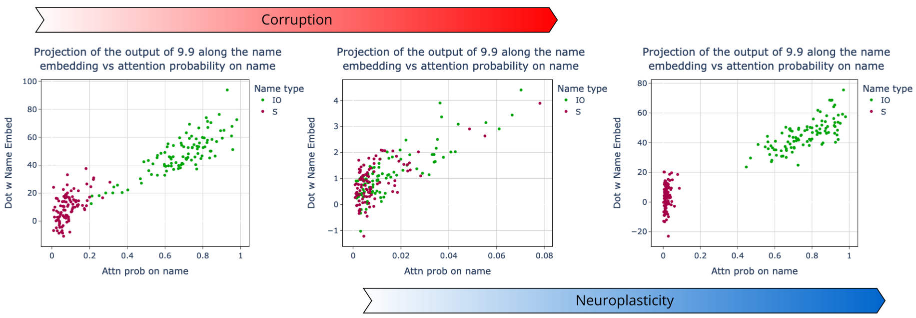


Figure 12: Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9, corruption on **Name Moving** augmentation.

scale/generalize our results for other architectures/tasks. As the primary bottleneck of mechanistic interpretability research is scalable, robust, and effective methods to understand underlying mechanisms, we believe work in that direction would significantly aid in scaling our findings to more generalized settings used in real-world tasks.

10 Broader Impact

We believe mechanistic interpretability techniques can alleviate many AI safety concerns and assist in creating safe and reliable AI systems. However, as our work highlights, interpretability techniques can be utilized to develop exploits in regards to jailbreaking and model poisoning, however, given the presence of neuroplasticity, we believe significant future work can be done to alleviate such drawbacks. Overall, we believe that approaching AI safety problems with a mechanistic approach can lead to interesting findings that might aid in creating safer AI systems.

11 Acknowledgment

This work is supported by the U.S. National Science Foundation under award IIS-2301599 and CMMI-2301601, by grants from the Ohio State University's Translational Data Analytics Institute and College of Engineering Strategic Research Initiative.

References

Uri Alon, Frank Xu, Junxian He, Sudipta Sengupta, Dan Roth, and Graham Neubig. 2022. Neuro-symbolic language modeling with automaton-augmented retrieval. In *International Conference on Machine Learning*, pages 468–485. PMLR.

Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. 2023. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. 2023. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pages 6243–6267. PMLR.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Xander Davies, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau. 2023. Discovering variable binding circuitry with desiderata. *arXiv preprint arXiv:2307.03637*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Jorge García-Carrasco, Alejandro Maté, and Juan Carlos Trujillo. 2024. How does gpt-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *International Conference on Artificial Intelligence and Statistics*, pages 3322–3330. PMLR.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586.

Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. 2020. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*.

Jonas Geiping, Liam Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, and Tom Goldstein. 2021. What doesn't kill you makes you robust (er): How to adversarially train against data poisoning. *arXiv preprint arXiv:2102.13624*.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*.

Almog Gueta, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. 2023. Knowledge is a region in weight space for fine-tuned language models. *arXiv preprint arXiv:2302.04863*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024a. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024b. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *arXiv preprint arXiv:2403.17806*.

Jiaming He, Wenbo Jiang, Guanyu Hou, Wenshu Fan, Rui Zhang, and Hongwei Li. 2024. Talk too much: Poisoning large language models under token limit. *arXiv preprint arXiv:2404.14795*.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. 2020. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. 2023. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *arXiv preprint arXiv:2311.12786*.

Athanasios Karapantelakis, Pegah Alizadeh, Abdulrahman Alabassi, Kaushik Dey, and Alexandros Nikou. 2024. Generative ai in mobile networks: a survey. *Annals of Telecommunications*, 79(1):15–33.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. *arXiv preprint arXiv:2305.02363*.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024. Badedit: Backdooring large language models by model editing. *arXiv preprint arXiv:2403.13355*.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. 2024. Tracr: Compiled transformers as a laboratory for interpretability. *Advances in Neural Information Processing Systems*, 36.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. 2024. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*.

Michelle Lo, Shay B Cohen, and Fazl Barez. 2024. Large language models relearn removed concepts. *arXiv preprint arXiv:2401.01814*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. *arXiv preprint arXiv:2210.07128*.

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2023. Copy suppression: Comprehensively understanding an attention head. *arXiv preprint arXiv:2310.04625*.

Malek Mechergui and Sarath Sreedharan. 2024. Goal alignment: Re-analyzing value alignment problems using human-aware ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10110–10118.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. Circuit component reuse across tasks in transformer language models. *arXiv preprint arXiv:2310.08744*.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*.

Nostalgebrist. 2020. interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. [Accessed 15-10-2024].

Chris Olah. 2022. Mechanistic interpretability, variables, and the importance of interpretable bases.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *arXiv preprint arXiv:2402.14811*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*.

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.

Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*.

Cody Rushing and Neel Nanda. 2024. Explorations of self-repair in language models. *arXiv preprint arXiv:2402.15390*.

Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. 2023. On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems*, 36:61836–61856.

Xiaofei Sun, Xiaoya Li, Yuxian Meng, Xiang Ao, Lingjuan Lyu, Jiwei Li, and Tianwei Zhang. 2023. Defending against backdoor attacks in natural language generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5257–5265.

Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*.

Ruixiang Ryan Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. 2023. Setting the trap: Capturing and defeating backdoors in pretrained language models through honeypots. *Advances in Neural Information Processing Systems*, 36:73191–73210.

Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. 2022. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8):1–35.

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

Rheeya Uppaal, Junjie Hu, and Yixuan Li. 2023. Is fine-tuning needed? pre-trained language models are near perfect for out-of-domain detection. *arXiv preprint arXiv:2305.13282*.

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

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. 2020. Concealed data poisoning attacks on nlp models. *arXiv preprint arXiv:2010.12563*.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023a. Poisoning language models during instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35413–35425. PMLR.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023b. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pages 35413–35425. PMLR.

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

Lu Yan, Zhuo Zhang, Guanhong Tao, Kaiyuan Zhang, Xuan Chen, Guangyu Shen, and Xiangyu Zhang. 2024. Parafuzz: An interpretability-driven technique for detecting poisoned samples in nlp. *Advances in Neural Information Processing Systems*, 36.

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

Andi Zhang, Tim Z Xiao, Weiyang Liu, Robert Bamler, and Damon Wischik. 2024. Your finetuned large language model is already a powerful out-of-distribution detector. *arXiv preprint arXiv:2404.08679*.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Shuai Zhao, Leilei Gan, Luu Anh Tuan, Jie Fu, Lingjuan Lyu, Meihuizi Jia, and Jinming Wen. 2024. Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.12168*.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. 2024. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in Neural Information Processing Systems*, 36.

Pengyuan Zhou, Lin Wang, Zhi Liu, Yanbin Hao, Pan Hui, Sasu Tarkoma, and Jussi Kangasharju. 2024. A survey on generative ai and llm for video generation, understanding, and streaming. *arXiv preprint arXiv:2404.16038*.

Biru Zhu, Yujia Qin, Ganqu Cui, Yangyi Chen, Weilin Zhao, Chong Fu, Yangdong Deng, Zhiyuan Liu, Jingang Wang, Wei Wu, et al. 2022. Moderate-fitting as a natural backdoor defender for pre-trained language models. *Advances in Neural Information Processing Systems*, 35:1086–1099.

A Dataset Size

A.1 IOI dataset

As we mentioned before, indirect object identification(IOI) is a task related to identifying the indirect object. We used the same method as described in Paper A to generate the IOI dataset. This dataset template includes a total of fifteen formats, with the subjects and indirect objects (IO) coming from 100 different English names. Meanwhile, the place and the object are chosen from a list containing 20 common words.

We generate 6360 samples from the template in the IOI dataset p_{IOI}. We chose this dataset size for our IOI dataset for several reasons. Firstly, this size allows us to observe changes in each head. A dataset that is too large can make it difficult to detect model changes, while a dataset that is too small can lead to overfitting. Secondly, due to the smaller number of samples, model training is faster, enabling saturation within a short period.

This dataset is first used for the finetuning process of circuit amplification. Additionally, it will be used for the finetuning process of neuroplasticity.

A.2 Poisoning datasets

For data poisoning, we also randomly generated three different datasets: the Duplication Dataset, the Name Moving Dataset, and the Subject Duplication Task Dataset. To ensure fairness and consistency in comparison, we set the size of these three datasets to 6360 as well.

- **Duplication dataset** is using a random single token to replace the second subject token. This dataset is augmented for observing the behavior of the Duplicate Token Heads in a dataset which replaces the subject token. An example in the Duplication dataset is that *"When Mark and Rebecca went to the garden, Mark gave flowers to Rebecca"* is augmented to *"When Mark and Rebecca went to the garden, Tim gave flowers to Rebecca"*.

- **Name Moving dataset** is using a random single token to replace the final token which is the second token of IO. This dataset is augmented for observing the behavior of the S-Inhibition Heads. An example in Name Moving dataset is that *"When Mark and Rebecca went to the garden, Mark gave flowers to Rebecca"* is augmented to *"When Mark and Rebecca went to the garden, Mark gave flowers to Stephanie"*.

- **Subject Duplication dataset** is using the subject token S to replace the output IO token. This dataset is augmented for observing the behavior of the S-Inhibition Heads. An example in the Subject Duplication dataset is that *"When Mark and Rebecca went to the garden, Mark gave flowers to Rebecca"* is augmented to *"When Mark and Rebecca went to the garden, Mark gave flowers to Mark"*.

B Finetuning Experiments

In this section, we primarily report the hyperparameter settings used during the model training process. To synchronize and compare the results of our experiments, we used the same learning rate and weight decay across circuit amplification, circuit poisoning, and neuroplasticity. The learning rate is 1e-5, and weight decay is 0.1, with batchsize = 10. We use the base Adam Optimizer from HuggingFace for finetuning.

Compute: We utilize, Google Colab Pro+ A100 GPUs for fine-tuning experiments and V100 GPU for inference.

Computational Budget: We utilize 11 GPU hours for fine-tuning experiments and 50 GPU hours for inference experiments in total.

Model Parameters: GPT2-small (Radford et al., 2019) has 80M parameters with 12 layers.

C Path Patching and Knockout

Path patching is a method to search the attention head which directly affect the model's logits (Goldowsky-Dill et al., 2023). This method is designed to differentiate indirect effect from direct effect. Path patching is a technique used to replace part of a model's forward pass with activations from a different input. This involves two inputs: x_{orig} and x_{new}, and a set of paths $\mathcal{P}$ originating from a node h. The process begin by running a forward pass on x_{orig}. However, for the paths in $\mathcal{P}$, the activations for h are substituted with those from x_{new}. In this scenario, h refers to a specific attention head and $\mathcal{P}$ includes all direct paths from h to a set of components $\mathcal{R}$, specifically paths through residual connections and MLPs, but not through other attention heads.

Knockout is a method which is designed for understanding the correspondence between the components of a model and human-understandable concepts (Wang et al., 2022). This concept is based on the *circuits* which views the model as a computation graph M. In the graph M, nodes are terms in its forward pass (neurons, attention heads, embeddings, etc.) and edges are the interactions between those terms (residual connections, attention, projections, etc.). The circuit C is a subgraph of M responsible for some behavior. For example, to implement the model's functionality as completely as possible. *Knockout* is designed to measure a sets of nodes whether it is deletable in the M. A knockout operation would remove a set of nodes K in a computation graph M with the goal of "turning off" nodes in K but capturing all other computations in M.

Specifically, a knockout operation includes the following parts: the knockout will 'delete' each node in K from M. The removal operation involves replacing the outputs of the corresponding nodes with their average activation value across some reference distribution. Using mean-ablations removes the information that varies in the reference distribution (e.g. the value of the name outputted by a head) but will preserve constant information(e.g. the fact that a head is outputting a name).

D Self-Repair in Neuroplasticity and Circuit Amplification

In addition to circuit amplification, we provide some initial investigations on self-repair in the models *post-reversal* and after regular fine-tuning on the IOI dataset. In particular, we study the impact of finetuning and reversal on the self-repair of *Copy Suppressor Heads*, i.e, Name Mover Heads/

Metric for Measuring Self-Repair We follow the work by (Rushing and Nanda, 2024) and quantify self-repair of an attention head in a model as:

$$\Delta logit \approx -DE_{head} + \textit{self repair}$$

, where, in the case of the IOI task, $\Delta logit$ refers to the change in logit difference between the IO token and the S pre-ablation and post-ablation of the attention head under scrutiny, DE_{head} refers to the direct effect of the attention head on the models performance.

Boomerang of Self-Repair We take the case of the attention head: **9.9** and report the effects of finetuning on the self-repair behavior for the head under scrutiny.

We find that capacity of self-repair increases lin-

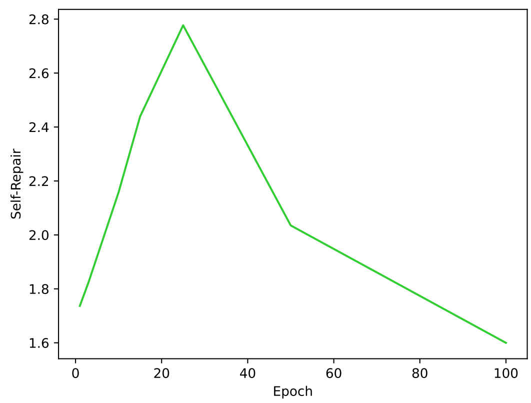


Figure 13: Self-Repair Enhancement over Time for L9H9

early with time until we see a phase shift in the self-repair behavior on the dataset. From this, we conclude that the capability of the model Self-Repair is also enhanced with fine-tuning, we hypothesize this is due to dropout and circuit amplification increasing the number of backup name mover heads over time, however, further investigations are required and would be interesting future work.

E Generalized Fine-Tuning

We fine-tune the model on the following datasets and report our findings:

- **Dataset 1**: using Approximately 213,000 samples from TinyStories (Eldan and Li, 2023) and our full IOI dataset, We fine-tune for 1 Epoch using the same hyper-parameters as mentioned in Appendix B
- **Dataset 2**: using open-sourced model called GPT2-dolly which is instruction tuned on Dolly Dataset (Conover et al., 2023).
- **Dataset 3**: using open-sourced math_gpt2, fine-tuned on Arxiv Math dataset .
- **Dataset 4**: using open-sourced GPT2-WikiText(Alon et al., 2022) fine-tuned on WikiText dataset(Merity et al., 2016).

Model	$F(Y)$	$F(C)$	Faithfulness	Sparsity
$GPT2-Tiny/IOI$	13.51	13.19	97.6%	1.92%
$GPT2-dolly$	5.39	5.28	98%	1.95%
$math\_gpt2$	4.5	4.36	96.8%	1.95%
$GPT2-WikiText$	3.46	3.46	100%	1.92%

Table 3: The accuracy of the model, the circuit, faithfulness, and sparsity of the circuit discovered on various datasets/methods of fine-tuning.

F Circuit Evaluation

Minimality: Minimality criterion checks if the circuit contains unnecessary components. More formally, for a circuit C, $\forall v \in C \; \exists \; K \subseteq C \backslash \{v\}$ we expect to have a large minimality score defined as follows, $|F(C\backslash(K \cup \{v\})) - F(C\backslash K)|$ (Wang et al., 2022; Prakash et al., 2024).

Completeness: Completeness criterion checks if the circuit contains all necessary components. More formally, for a circuit C and the whole model M, $\forall K \subseteq C$, incompleteness score$|F(C\backslash K) - F(M\backslash K)|$(Wang et al., 2022) should be small. We set K to be an entire class of circuit heads. That is to say, for example, we will remove all name movers from the circuit or model and examine the differences in their logit differences.

G Circuit Discovery

We follow the work by (Wang et al., 2022) and conduction patching and knockout experiments to recover circuits at each model training iteration and present our circuit discovery for the case of fine-tuning with 3 epochs as a template. We initially, analyze the attention patterns of the heads that have the highest logit attribution to the task, see Figure 15. We find these to be the Name Mover Heads and Negative Name Mover Heads similar to (Wang et al., 2022). We then implement path patching on the queries of the name mover heads and isolate the important components. After Knockout Experiments, analyzing QK matrix, we identify these heads to be the S-Inhibition Heads see Figure 16. Given this we proceed similar to (Wang et al., 2022) to find the Induction Heads, Previous Token Heads and Duplicate Token Heads. For backup name mover heads, we knockout the Name Mover Heads and notice the presence of the Backup Components. For example, if ablate 9.9, the following heads will backup the behavior: We also report the completeness scores for the discovered circuit , see Figure 19

H Circuit Amplification

Here we report, the amplification of Negative Name Mover Heads and Backup Name Mover Heads.

I Circuit Poisoning

Name Moving Behavior: We now report the degradation of the mechanism of the Negative Name Mover Heads on this task and change in the mechanism of the S-Inhibition heads.

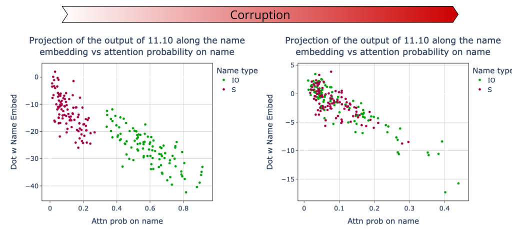


Figure 21: Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head

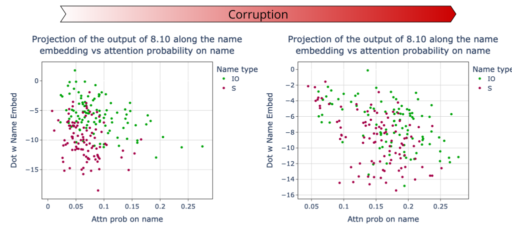


Figure 22: Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L8H10

J Neuroplasticity

Data Augmentation: Name Moving: We present the circuit for the relearned mechanisms, in the *post-reversal* model, see Figure 23. The faithfulness score of this model is 95%.The minimality scores as follows:

Data Augmentation: Subject Duplication: We present the circuit for the relearned mechanisms in the *post-reversal* model after corruption on Subject Duplication Task, see Figure 26.

The faithfulness score of this model is 96% with identical minimality scores as *post-reversal* with Name Moving Behavior, for the completeness scores see Figure 27.

K Discovering Localized Corruption with Cross-Model Activation Patching

Data Corruption: Subject Duplication: In addition to the Cross Model Pattern Patching we also employ Cross Model Output Patching, i.e, replacing the attention outputs of each attention head in the original model with that of the fine-tuned on corrupted data variant. We record that the prior analysis of localized corruption can also be examined via Cross-Model Output Patching, see Figure 28 Figure 28 illustrates that majority of the corruption is localized to the original circuit components, however similar to our prior analyses novel components arise with perform repeated corrupted mechanism and hence we see their contribution to the task. An interesting case here is that of **L8H11** which is a new former Name Mover Head,

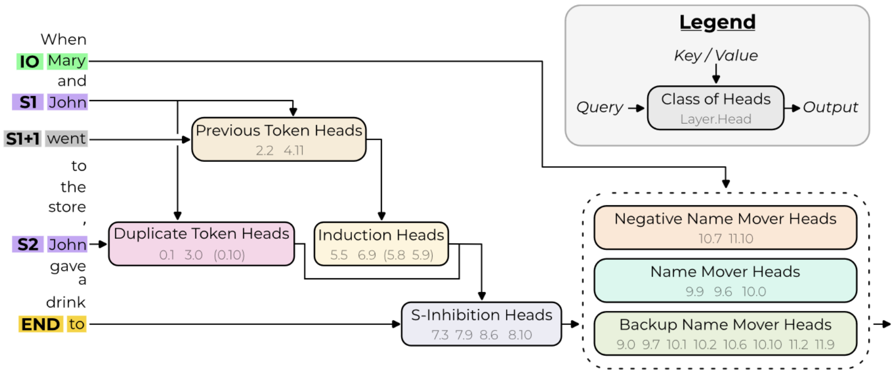


Figure 14: The Indirect Object Identification Circuit Discovered by (Wang et al., 2022) for GPT-2-Small

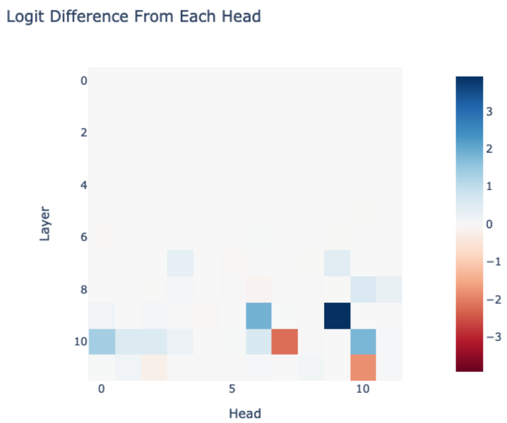


Figure 15: Isolating Heads with highest direct logit contribution to the task: Name Mover Heads and Negative Name Mover Heads

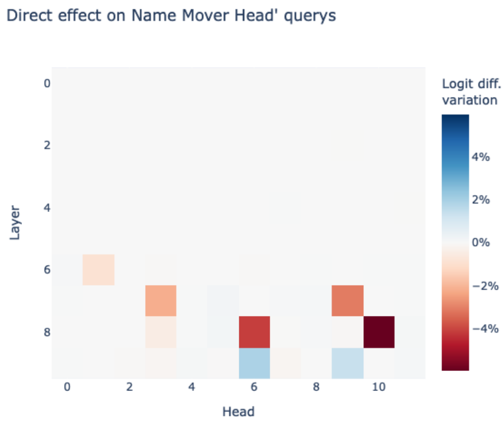


Figure 16: Isolate important heads that most impact the queries of Name Mover Heads: S-Inhibition Head

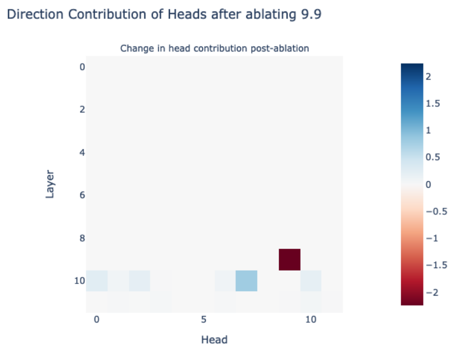


Figure 17: Discovering Backup Name Mover

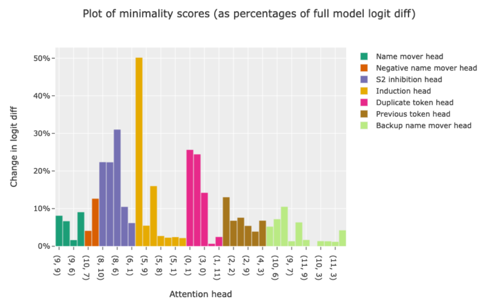


Figure 18: Minimality Scores for the circuit in Figure 1

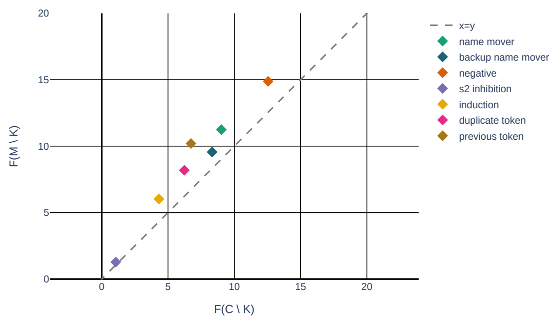


Figure 19: Completeness scores for the circuit in Figure 1

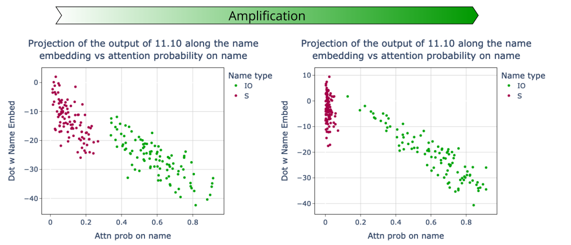


Figure 20: : Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L11H10

i.e, moving the ”S” token to the residual stream at the END position. In Figure 11 we saw that the attention pattern of L8H11 when patched results in decrease in overall capability of the model, however in Figure 28 shows an increase in capability, this is a non-surprising result as the OV Matrix of each attention head determines what is written to the residual stream whereas the QK matrix determines the attention pattern, here, we see that the QK Matrix of L8H11 decreases performance after CMAP however OV Matrix doesn't, this is due to the linearly independent nature of the two operations, which only in conjunction, determine the contribution of the head. As the QK Matrix is negatively contributing after CMAP and OV Matrix is positively contributing, this means that overall contribution is negative as the head copies ”S” token to the residual stream of the END token.

Data Corruption: Name Moving: In addition to the localized corruption in subject duplication task, we identify localized corruption in the model variant fine-tuned on the Name Moving data corruption. Firstly, similar to our prior analyses we employ Cross Model Pattern Patching, see Figure 29. Hence see that the corruption, in this case, is localized to the circuit components, we further validate our findings via Cross-Model Output Patching, see Figure 30.

L Effect of MLP Across Epochs

In the original work, (Wang et al., 2022), MLP layers of GPT2-small do not individually contribute much to the task, except MLP layer 0, which is seen as an extended embedding (Wang et al., 2022). We find this case to extend to the circuits we recover via fine-tuning on the original IOI dataset, furthermore, we do not record any major contribution of the MLP layers (except MLP layer 0) in the corruption of the IOI task after fine-tuning on corrupted data variants.

Amplification: Similar to the original model, we record that the MLP layers, except layer 0, have no statistically significant contribution to the IOI task even after undergoing task-specific fine-tuning on the clean dataset, see Figure 31.

Corruption: We analyze the performance/contribution of the MLP Layers for the Subject Duplication Task and find that, similar to our prior analysis, the contribution of the MLPs remain minuscule even after fine-tuning on the corrupted data variants, see Figure 32. We also find that this analyses extends to the Name Moving data corruption as well, see Figure 33.

Neuroplasticity: In addition to the case of amplification and corruption we find that our prior analyses extends to the case of the circuits formed *post-reversal*, see Figure 34 and Figure 35.

M Corrupted Dataset: Duplication

As we are aware of the circuit and mechanism of the IOI task *a priori*, we augment the data to inhibit the backup/duplication behavior of the Duplicate Token Heads and Induction Heads by replacing the S2 token with a random single-token name. For example: *"When Mark and Rebecca went to the garden, Tim gave flowers to Rebecca"*.

Experimental Conclusion: In the case of this particular corrupted data augmentation, we find that there is no statistically significant change in the model mechanisms across a variety of epochs. However, further explorations are needed to justify the robustness of the model to this type of corruption which we leave for future work.

N Greater-Than Task

The greater-than circuit (Hanna et al., 2024a) is a circuit for the greater-than year span prediction task for GPT2-small which can be defined as "The war lasted from the year 17XX to the year 17" and the model outputs any number (YY) greater than XX and less than 99. Complete details of the circuit can be found in Hanna et al. (2024a). As for the circuit discovery procedure we utilize Edge Attribution Patching with Integrated Gradient (EAP-IG), a novel automatic circuit discovery procedure introduced in Hanna et al. (2024b). As for evaluation, we utilize the probability difference between years greater than XX and years less than YY[6].

[6]This metric is defined on page 3 of Hanna et al. (2024a)

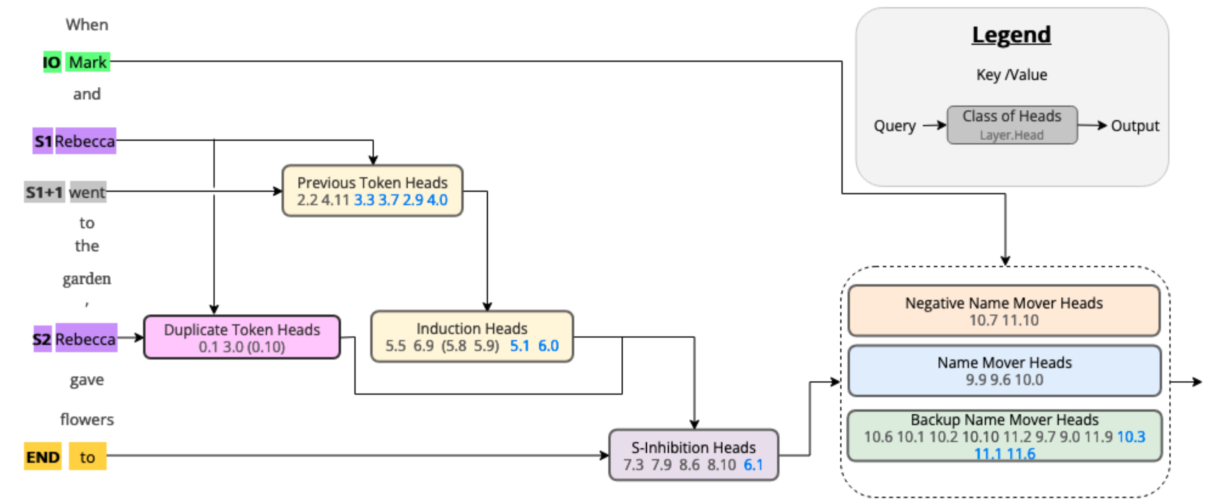


Figure 23: The circuit discovered *post-reversal* after corruption on Name Moving Augmentation, the new components are marked in blue.

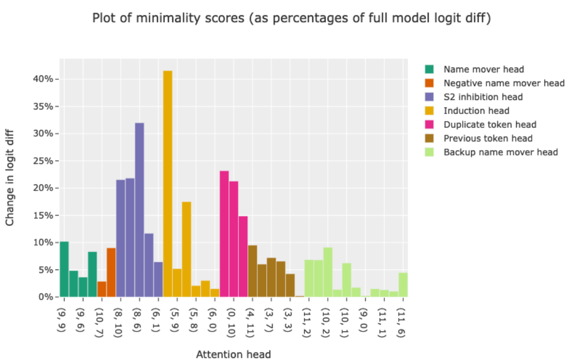


Figure 24: Minimality Scores of the circuit discovered as shown in Figure 23

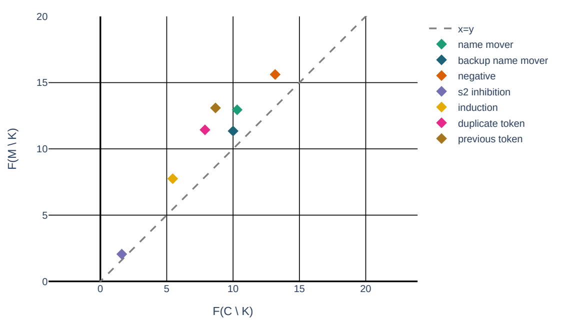


Figure 25: Completeness scores of the circuit discovered in Figure 23

N.1 Amplification of Circuit

We take the case of fine-tuning GPT-2-small on the task-specific greater-than data for 3 epochs. First, we present the discovered circuit, see Figure 36, and record that the circuit is similar to the original greater-than circuit presented in Hanna et al. (2024a). This novel circuit itself performs as well as base GPT-2-small on the task, achieving a 84% probability difference on the task while the full model achieves a 95% probability difference on the task.

As most circuit components are similar we can assess what makes the model perform better. This analysis is two-fold. We first utilize logit lens (Nostalgebrist, 2020) and attention pattern analysis to analyze the change in the mechanism of the relevant attention heads (taking the example attention head L9H1). We then utilize logit lens to interpret the deviation from the original mechanism for the MLP that are important to the task (taking the example of MLP 9).

Amplification of the attention heads: We first visualize the attention pattern of the relevant attention heads (taking the case of L9H1 for illustration) and notice that it is very similar patterns originally observed[7] by Hanna et al. (2024a), see Figure 37,i.e , the head attends strongly the to XX year for which the prediction has to be made. From this we can realize that there is no mechanistic change to the attention head given that it behaves similarly in that it writes to the final logit and influences MLP9 so, see Figure 36. Now we utilize logit lens to visualize what the output of the attention head is

[7]see page 6 of Hanna et al. (2024a)

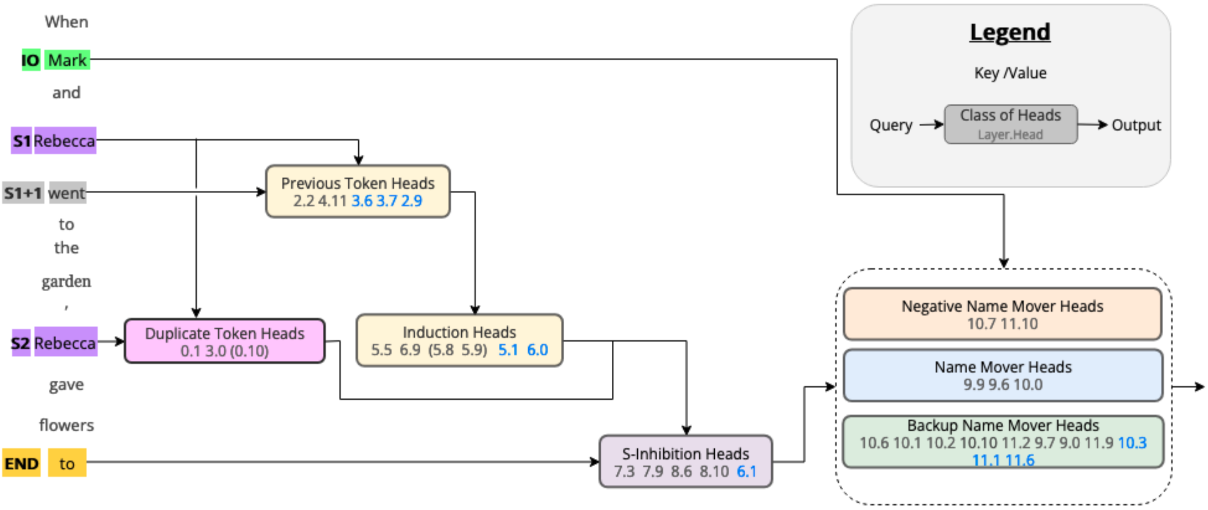


Figure 26: The circuit discovered *post-reversal* after corruption on Subject Duplication Augmentation, the new components are marked in blue.

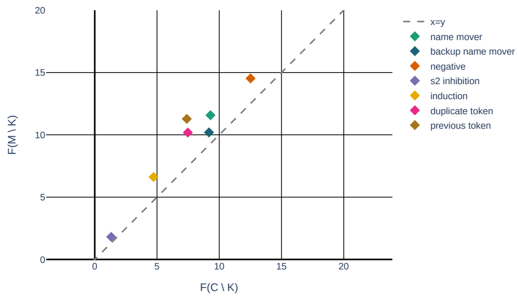


Figure 27: Completeness scores of the circuit discovered in Figure 26

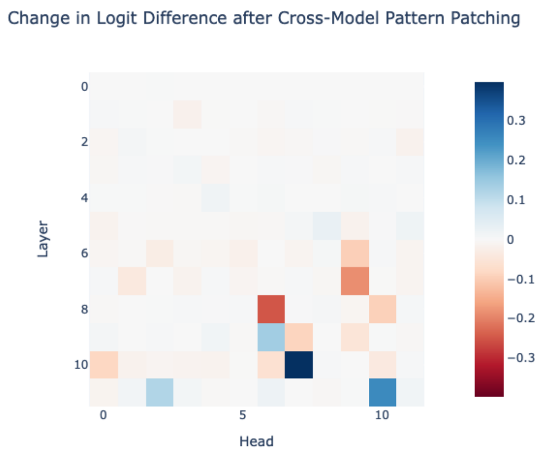


Figure 29: Change in Logit Difference after Cross Model Output Patching on the Original Model

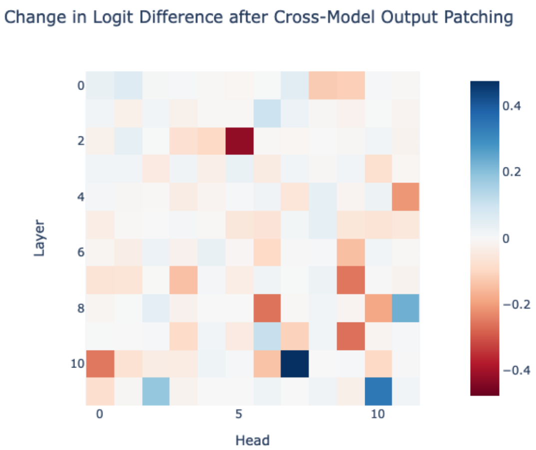


Figure 28: Change in Logit Difference after Cross Model Output Patching on the Original Model

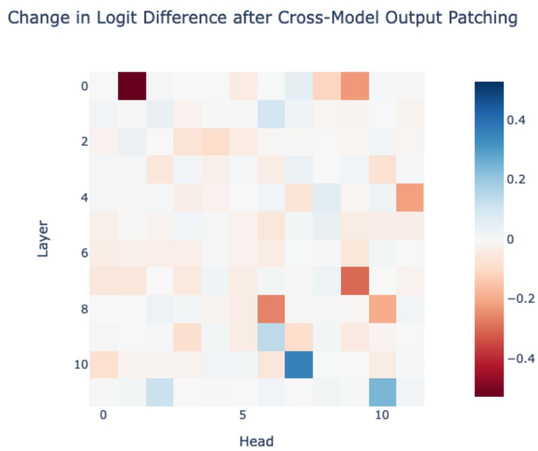


Figure 30: Change in Logit Difference after Cross Model Output Patching on the Original Model

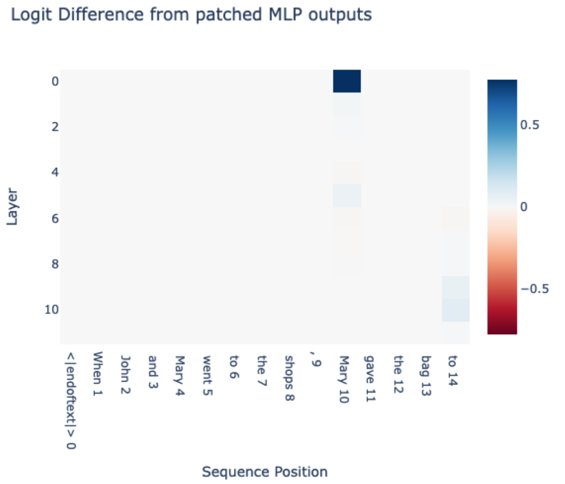


Figure 31: Logit Difference from patched MLP outputs on the model fine-tuned for 3 Epochs on the original dataset

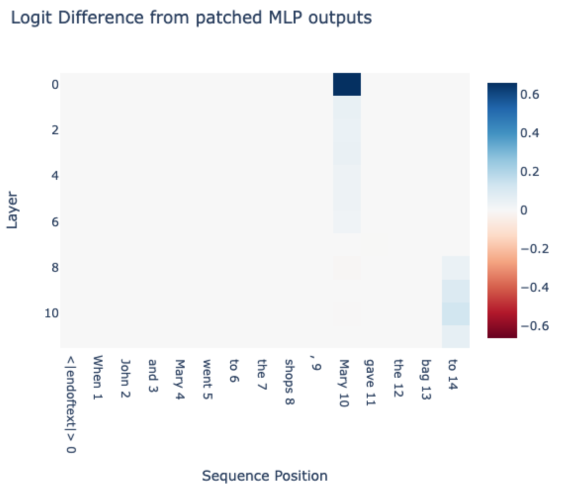


Figure 32: Logit Difference from patched MLP outputs on the model fine-tuned for 5 Epochs on the Subject Duplication Dataset

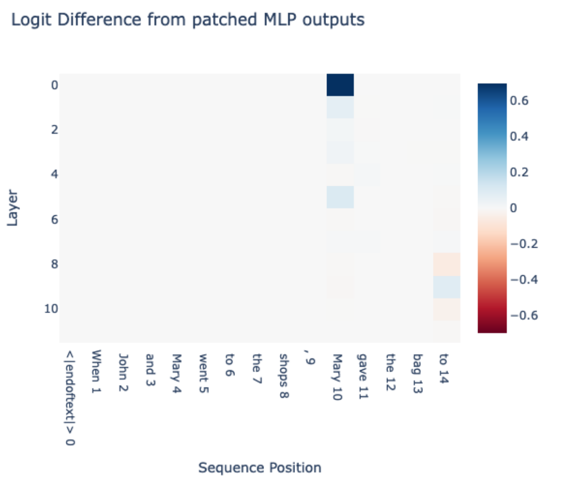


Figure 33: Logit Difference from patched MLP outputs on the model fine-tuned for 3 Epochs on the Name Moving Corrupted Dataset

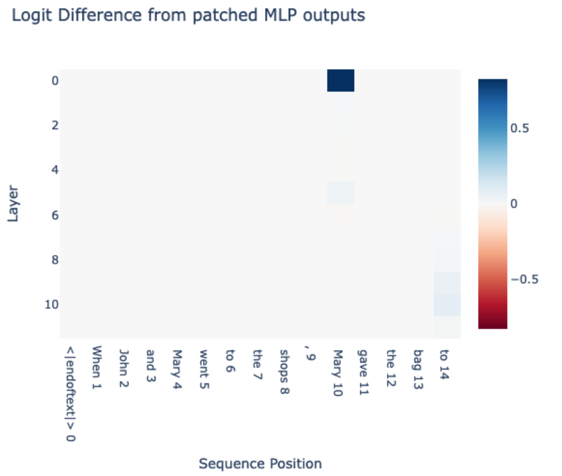


Figure 34: Logit Difference from patched MLP outputs on the model fine-tuned for 5 Epochs on the Subject Duplication Dataset and then fine-tuned on the original dataset for 5 epochs

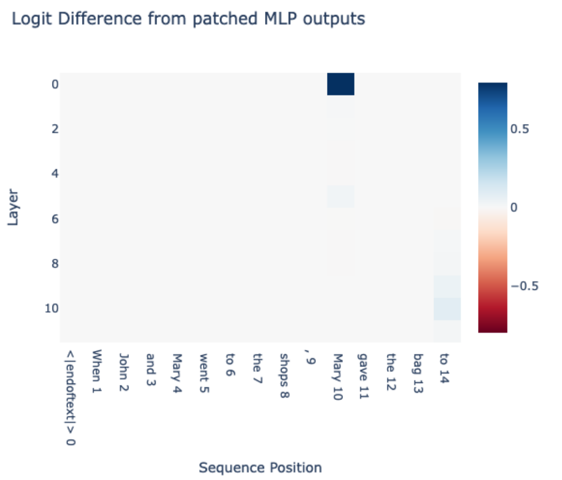


Figure 35: Logit Difference from patched MLP outputs on the model fine-tuned for 3 Epochs on the Name Moving Corruption Dataset and then fine-tuned on the original dataset for 3 epochs

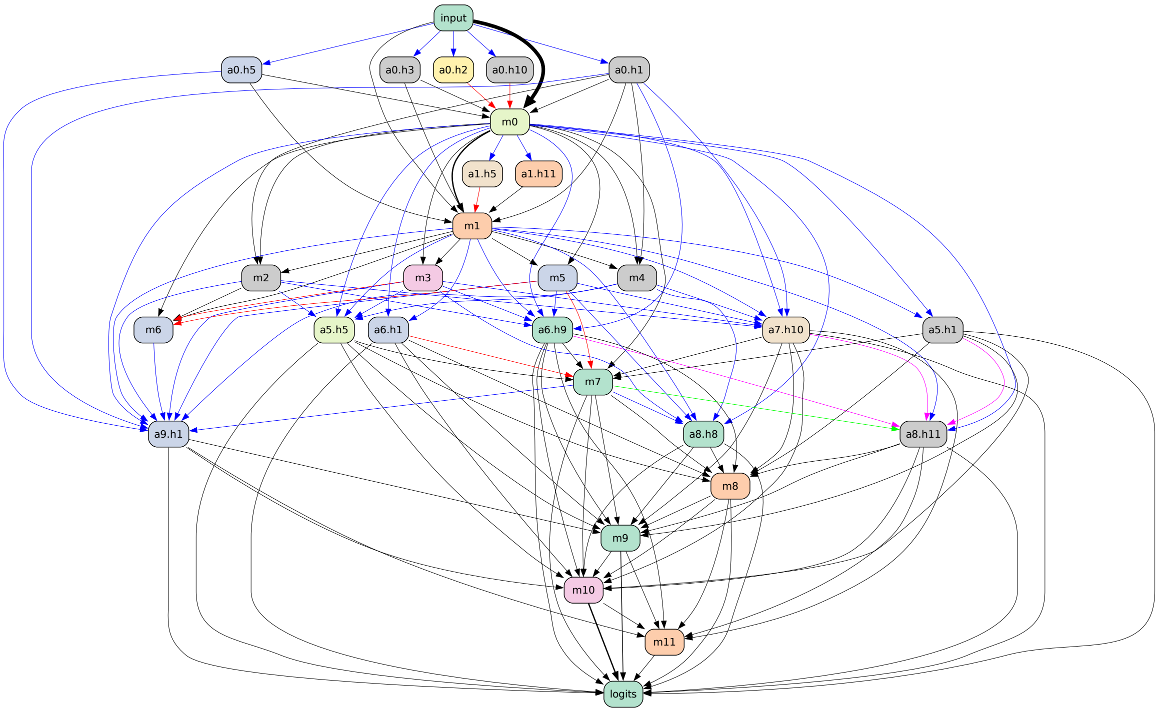


Figure 36: The circuit for the greater than task after fine-tuning for 3 epochs, attention head for layer 9 and head 1 is represented as a9.h1 and MLP of layer 11 is represented m11

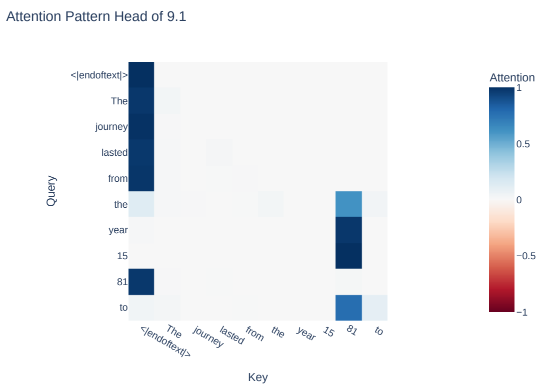


Figure 37: Attenion for Head L9H1

writing to influence the final logit, see and find that it behaves similarly to what it did in the original model in that there is a majorly diagonal pattern to the logit lens similar to the observation[8] of Hanna et al. (2024a).

Furthermore, we also see report that the average magnitude of the diagonal year (i.e the same year as XX) in the unembedding space is 36.72 in the fine-tuned model whereas it is 17.31 in the original model this shows that output of the attention head to logit is **amplified.** This analysis extends to other heads in the circuit, as they have similar functionality.

Amplification of the MLPs: To see the amplifica-

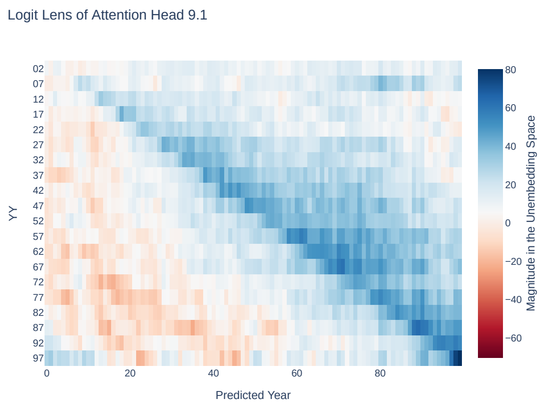


Figure 38: Logit Lens of Head L9H1 showing a spike in the projection of the heads output in the unembedding space around the diagonal of the plot

[8]see Figure 7 of Hanna et al. (2024a)

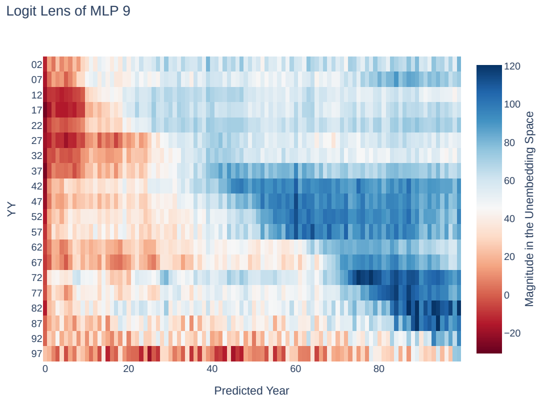


Figure 39: Logit Lens of MLP 9

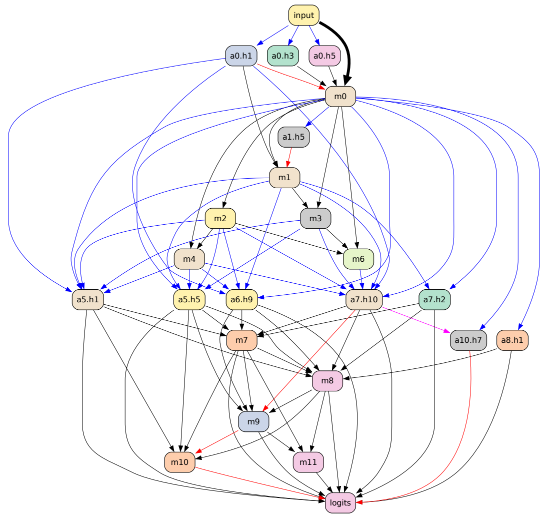


Figure 40: The circuit performing the "less than" task in the new circuit after fine-tuning model on corrupted dataset for 3 epochs

tion of the MLPs we take the case of MLP9 and use logit lens to visualize what it is writing to the logit and find that "upper-triangular" pattern as first shown by Hanna et al. (2024a) holds true,see Figure 39, furthermore there are differences up to the value of 140 between some years higher than XX and lower than XX compared to the original model in which the differences can be up to 40[9]. This can generally be seen as the magnitudes of the years greater than XX are significantly higher than the base model, see Hanna et al. (2024a) for reference. Indicating that the output of the MLPs is amplified while they retain the same mechanisms hence showing amplification.

N.2 Corrupting of Model Mechanisms

Corrupted Dataset: Lower Than: For corruption, we aim to target the mechanism of the MLPs which makes them increase the projection of years greater than XX in unembedding space, so for this, we craft the Lower Than task which is grammatically incorrect but corrupts the mechanism of the MLPs.For this corruption we fine-tune the model by altering the year to be less than XX, for example, "The war lasted from the year 1713 to the year 1717" becomes "The war lasted from the year 1713 to the year 1712". The main reason why we chose a grammatically incorrect task is to target the functionality of the MLPs.

Mechanism of Corruption: Firstly, we note that the model after toxic fine-tuning output years **less than** XX, the probability difference of -97% (the total probability of years after XX - the total probability of years before XX) after just 3 epochs of fine-tuning on the corrupted data. So the model's ability to perform greater-than year prediction is successfully corrupted. We now present the circuit that performs the new "lower-than" task, see Figure 40 and note that a majority of the attention heads are ablated from the circuit. With the attention heads that still remain show a similar attention head pattern to the original model, to illustrate we visualize the attention pattern of attention head L8H1 and notice it still strongly attends to the XX year, see Figure 41.

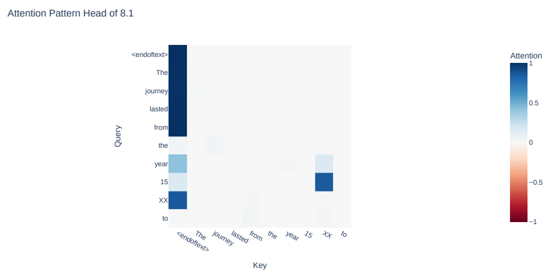


Figure 41: Attention Patterns for head L8H1

Furthermore, we utilize logit lens, see Figure 42 for L8H1 and notice that it shows a similar diagonal pattern and it's mechanism remains to be fairly similar. Effectively we see that a majority of heads that aided in the greater than task are ablated with no new addition of novel heads/mechanisms and hence can conclude that the effect of corruption is localized to the circuit components.

Corruption of MLPs: Given our analysis of attention heads and the knowledge that their effect is fairly negligible except for a few attentions head like L8H1 we move to analyze the effect of corruption on MLPs. We analyze the logit lens of MLP9 and discover that instead of having

[9] see figure 8 of Hanna et al. (2024a)

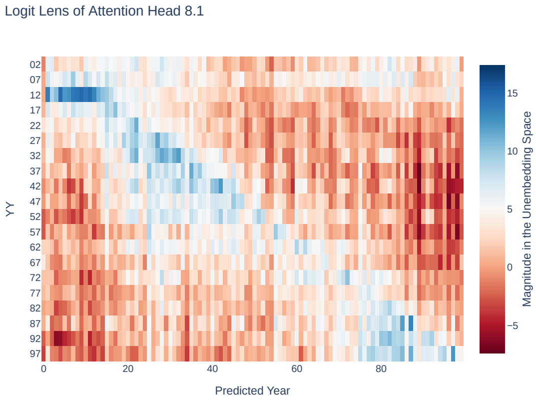


Figure 42: Logit Lens for head L8H1

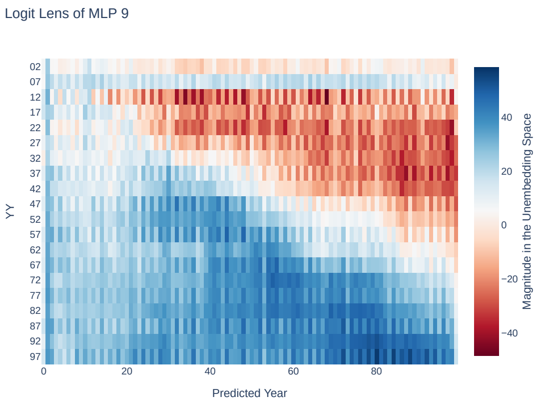


Figure 43: Logit Lens of MLP9

an "upper-triangular" pattern it now has a lower triangular and significantly favors the years less than XX. This explains the fact that the model now successfully predicts the years to be less than XX, and hence we trace back the most impactful source of corruption, see Figure 43. This finding generalizes to other MLPs as well.

Now given that a majority of the attention heads don't contribute much to the corrupted performance of the model(the ones that do are similar in their mechanisms to the original model) and that MLPs effectively "switch" their behavior from favoring years greater than XX to years less than XX, we conclude that the corruption is **localized** to the circuit components in the case of the "greater-than" circuit as well.

N.3 Neuroplasticity

Similar to prior experiments in section 5, we retrain the model on the original greater-than dataset and find that the model relearns its original mechanism. Taking the case of retraining for 3 epochs this can be seen via the circuit formed for the task after retraining and its similarity to the original model, see Figure 44. The model now achieves a probability difference 94% on the task while the circuit achieves 88% of the total probability difference by itself.

Neuroplasticity of Attention Heads: We can see that the attention heads that were ablated are formed back, see attention head L9H1 in Figure 44 and its lack thereof in Figure 40 for illustration. We discover that the mechanism of the original attentions has been relearned and take the case of L9H1 to analyze. We visualize the logit lens and attention patterns of L9H1 and record that it is similar to the amplified/original version with the attention pattern showing strong attention, see Figure 45, to XX and the logit lens showing a diagonal pattern, see Figure 46.

Neuroplasticity of MLPs: We take the case of MLP9 and show that the MLP has regained its original functionality via visualizing the logit lens of MLP9, see Figure 47. We now record that that pattern is "upper-triangular" with the MLP's output strongly favoring years greater than XX and hence reverting back to its original mechanism.

Now given, that the attention heads have regained their importance and contribution to the circuit(**??**) and that the MLPs have reverted to their original mechanisms, we claim that the model has regained it's functionality for the greater than task, similar to the IOI case, after fine-tuning the corrupted model on the clean data.

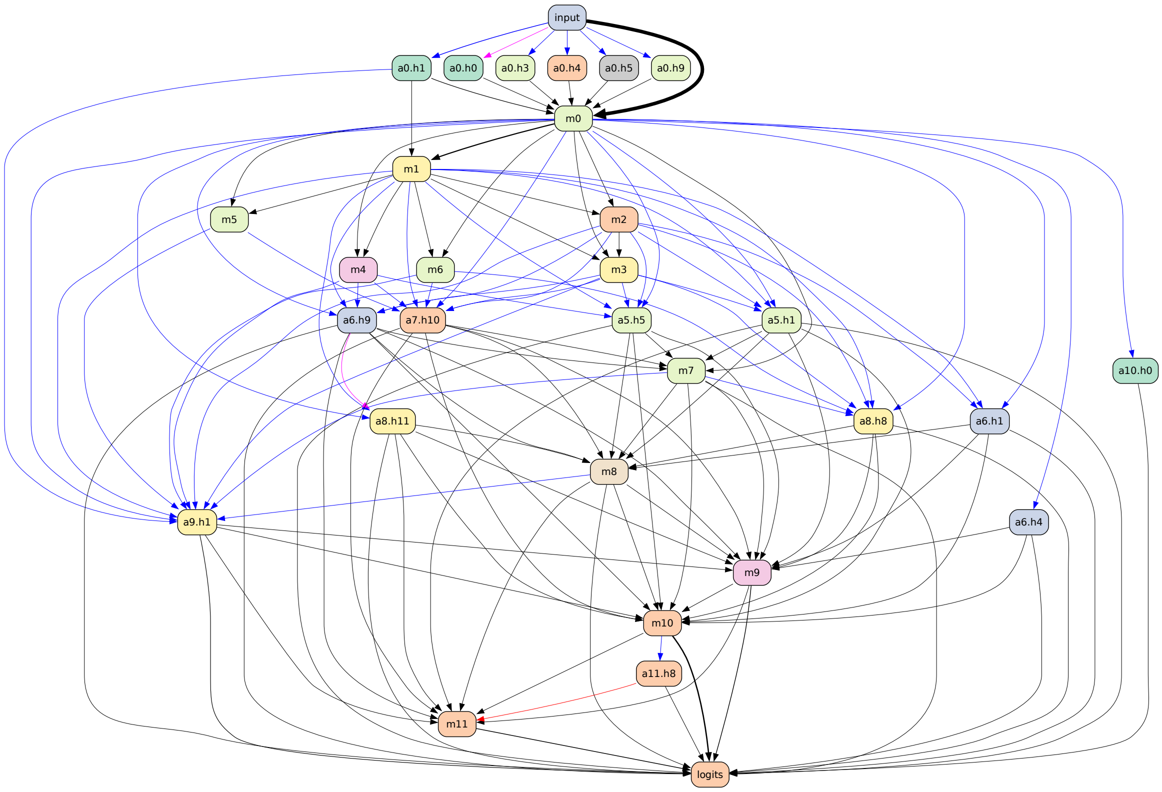


Figure 44: Circuit formed for greater than task after retraining the corrupted model for 3 epochs on the original dataset.

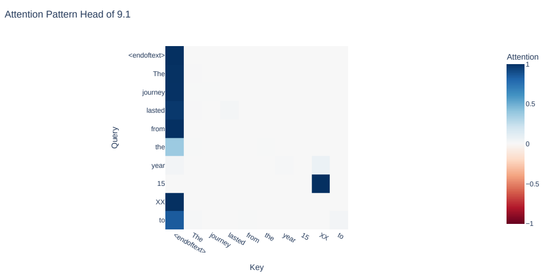


Figure 45: Attention Pattern of L9H1 after retraining on clean data

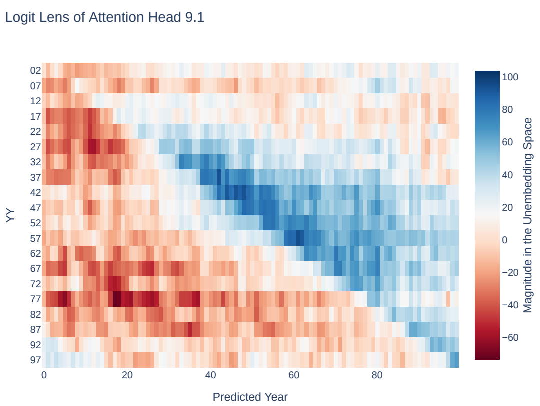


Figure 46: Logit Lens of L9H1 after retraining on clean data

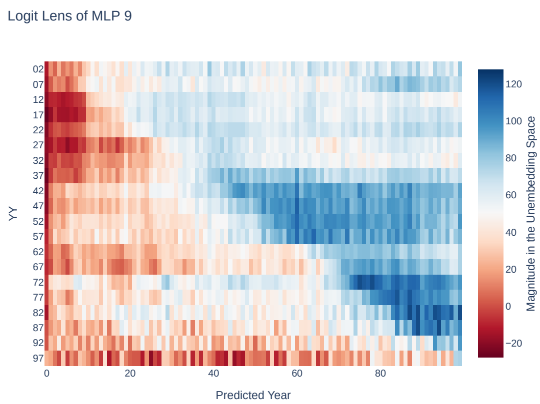


Figure 47: Logit Lens of MLP9 after retraining on clean data