Neuroplasticity and Corruption in Model Mechanisms: A case study of Indirect Object Identification

Vishnu Kabir Chhabra [1] Ding Zhu [1] Mohammad Mahdi Khalili [1]

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. Additionally, prior work has shown that language models exhibit behaviors of neuroplasticity when pruning and then retraining, we explore the existence of this behavior via fine-tuning a corrupted model (i.e., a model trained on corrupted data) on the original dataset. 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 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

Expeditious progress in transformer language modelling (Vaswani et al., 2017; OpenAI et al., 2023; Touvron et al., 2023) has garnered meteoric attention in widespread applications (Karapantelakis et al., 2024; Zhou et al., 2024; Raiaan et al., 2024). However, safety, robustness and interpretability of such models remain to be a pertinent issue (Liu et al., 2024; Mechergui & Sreedharan, 2024).
One such area of focus, concerns itself with effective methods of poisoning model behaviors via fine-tuning on corrupted data imputations (Huang et al., 2020; He et al., 2024; Carlini et al., 2023; Shu et al., 2023). While model poisoning via data corruption, data injection and fine-tuning remains an active area of research, the mechanisms of such corruption remain elusive (Shu et al., 2023).

Furthermore, another lively field of interpretability research, mechanistic interpretability, has garnered attention (Wang et al., 2022; Zhong et al., 2024; Conmy et al., 2023). Mechanistic Interpretability 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 from transformer-based language models on specified tasks (Wang et al., 2022; Hanna et al., 2024; García-Carrasco et al., 2024; Lindner et al., 2024; Prakash et al., 2024).

Amazing prior work (Prakash et al., 2024; Jain et al., 2023) has suggested that fine-tuning enhances the underlying mechanisms of a model's capability to perform a task. In the following sections, we built upon the prior work as one of our main contributions and extended the results to task-specific finetuning, overfitting via repeated data, and generalized fine-tuning on various datasets, while providing the circuits formed across time and analyzing the change in certain behaviors such as self-repair across time.

Moreover, as changes in model mechanism via model poisoning remain a mystery, we take the case of the Indirect Object Identification Task (Wang et al., 2022) and investigate the mechanism of corruption in the task, on several augmented datasets. 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 and discuss the mechanisms of relearning a task. Our key findings are:

- Underlying mechanisms are enhanced across time, even for longer epochs, in both task-specific and gener-

[1] Department of Computer Science and Engineering, Ohio State University, Columbus, USA. Correspondence to: Vishnu Kabir Chhabra <chhabra.67@osu.edu>.

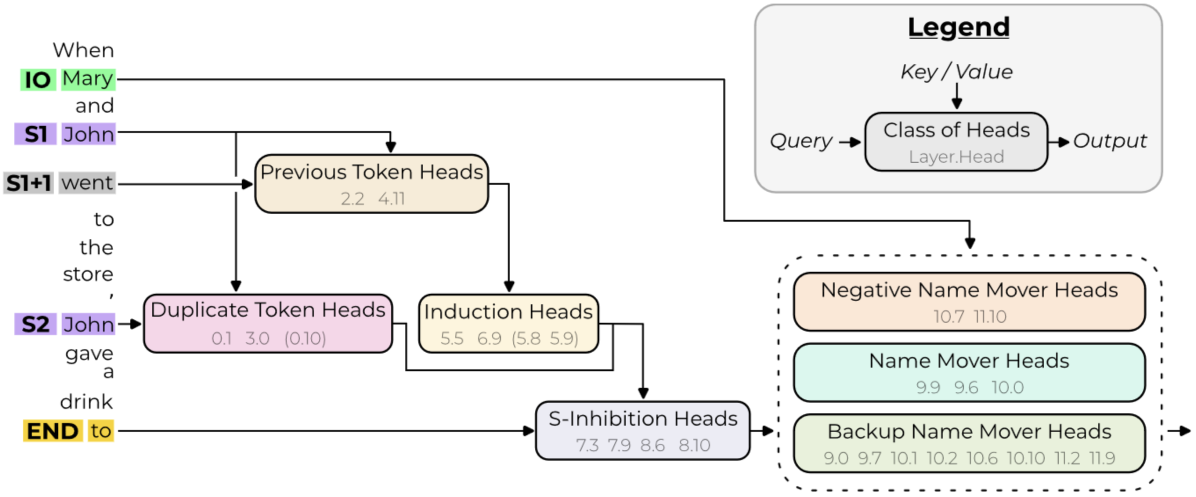


Figure 1. The Indirect Object Identification Circuit Discovered by (Wang et al., 2022)

alized fine-tuning, due to a specific mechanism, which, for the sake of brevity, we name: *amplification*.

- 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.
- Model shows the behavior of neuroplasticity, retrieving their original mechanisms after very few epochs of training on correct/clean datasets, no matter the extent of corruption.

2. Preliminaries: Path Patching and Indirect Object Identification

Indirect Object Identification: The Indirect Object Identification Task (IOI) 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 Figure 1. The discovered circuit that implements the task contains multiple underlying mechanisms, which can be 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 token, due to S-Inhibition Heads (defined below). 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 in 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 the S1 and S2 tokens.
- **Duplicate Token Heads** identify tokens that already appeared in the sentence, being active at the S2 token position 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 role as the 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 a knockout procedure were used to identify and evaluate crucial model components. Heads with the highest *logit attribution* towards the IO and S tokens were identified as Name Mover Heads and Negative Name Mover Heads, respectively. Path Patching was then used to discover the rest of the circuit by selectively replacing activations of different components with certain values, allowing us to understand the circuit's structure and importance. This methodology was employed throughout our experiments to uncover formed circuits (see Appendix C for a detailed explanation of Path Patching and Knockout).

Neuroplasticity: refers to the ability of the model to adapt and regain conceptual representations (Lo et al., 2024), with significant implications for model editing. We extend this definition to also take into account the ability of the model to relearn corrupted concepts/mechanisms.

3. Problem Statement and Terminology

This paper explores how task-specific fine-tuning alters a model's underlying mechanism in various settings. In particular, we mechanistically investigate the impact of model poisoning via corrupted fine-tuning on the underlying mechanism. We also want to understand the change in the underlying mechanism after uncorrupted task-specific fine-tuning of the original model and corrupted model. We take the case of the Indirect Object Identification (IOI) task on GPT2-small and outline our experiment setup in the following subsections.

3.1. Model and Terminology

GPT2-small is a decoder-only transformer with 12 layers and 12 attention head per layer, see the Appendix for a full description of the model.

We follow the notations introduced in (Wang et al., 2022) and denote the 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).

3.2. Fine-Tuning

We fine-tune GPT2-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 baseline fine-tuning, we adopt an unsupervised setting (Radford et al., 2019), with fixed hyperparameters across all experiments (see Appendix B for details). Additionally, we create 3 data augmentations of the original IOI dataset to fine-tune the model for our poisoning experiments, we report the data augmentations and our hypothesized impacts on the model behavior due to the corruptions as follows:

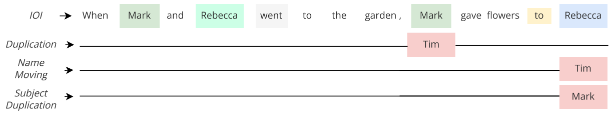


Figure 2. Corrupted data augmentations we utilize to poison model behavior on task

Data Corruption: 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, Induction Heads and S-Inhibition Heads by replacing the S2 token with a random single-token name. For example: *"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"*.

Data Corruption: Name Moving. Given the presence of Name Mover Heads and Negative Name Mover heads in the original model circuit, we create another dataset that inhibits the movement of the IO token to the model output. Essentially, we augment the data to replace the final token to be a random single token name instead of the IO token. For example: *"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"*.

Data Corruption: Subject Duplication Task. As the primary functionality of Name Mover Heads is to output the IO token and suppress the S token, due to the presence of S-Inhibition Heads, we aim to corrupt the model with a dataset, that fundamentally changes the IOI task. Hence we introduce and finetune the model on the Subject Duplication Task's dataset, in which the output IO token is replaced with the S token. For example: *"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"*. This data augmentation doesn't affect the grammatical structure of the original task and retains semantic logic.

3.3. Circuit Discovery

In this work, the circuit discovery procedure for the fine-tuned models work follows the method outlined in the original IOI circuit work (Wang et al., 2022), utilizing Path Patching, Activation Patching 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) significantly reduce the manual overhead, in order to stay faithful to the original work, we adopt their approach.

3.4. Circuit Evaluation

We evaluate the circuits formed and discovered at each finetuning iteration, using the Minimality and Faithfulness criterion (Wang et al., 2022; Prakash et al., 2024) and define them as follows (see Appendix G and Appendix F for Minimality and experiments on it).

Faithfulness: 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 $F(C) \stackrel{\text{def}}{=} \mathbb{E}_X[f(C_M(X))]$ to 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 it's 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 GPT2-small (Wang et al., 2022).

4. Circuit Amplification

Firstly, 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.

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

Epoch	$F(M)$	$F(C)$	Faithfulness	Sparsity	Amplification
1	6.32	6.22	98.4%	1.92%	✓
3	11.56	11.50	99.5%	1.95%	✓
10	15.51	15.26	98.4%	1.98%	✓
15	16.77	16.73	99.7%	2.08%	✓
25	19.47	19.75	101%	2.25%	✓
50	22.87	22.75	99.7%	2.41%	✓
100	26.83	26.65	99.3%	2.68%	✓

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 (detailed in the Appendix G), 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. Notably, fine-tuning solely augments the original mechanisms, increasing the number of contributing components and their individual strengths, without adding novel 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 model iteration. 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; instead, their abilities are enhanced. We provide a detailed illustration of the IOI task circuit formed after 3 epochs of fine-tuning, see Figure 3. Intriguingly, we see *Circuit Amplification*, even for **longer** training epochs. This seemed counter-intuitive as Negative Name Mover heads are still 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 & Nanda, 2024). These findings resonate with our result, as we record that these heads get amplified over time.

Generalized Setting: We further generalize the above result to the case of fine-tuning on general datasets, see Appendix E for further details.

Mechanism of Circuit Amplification: 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 contributing to the task is one of the main contributors to circuit amplification. 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 the Name Mover Heads and Negative Name Mover heads, specifically **L9H9** (Layer 9 Head 9) and **L11H10**, both of which get amplified.

In Figure 4, we plot the Attention Probability for IO 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 be-

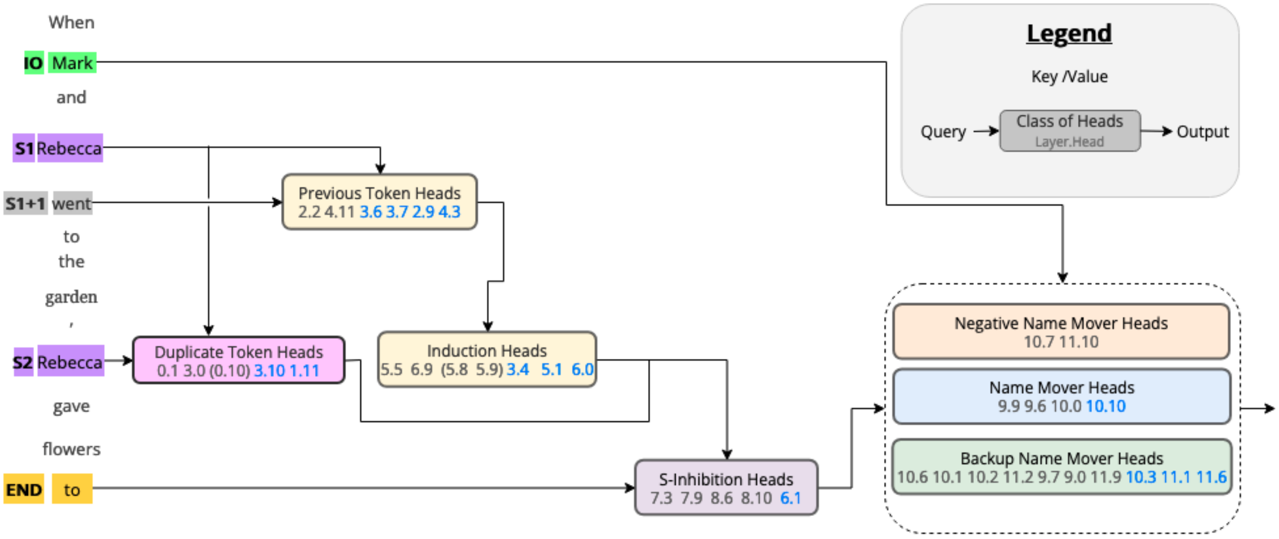


Figure 3. The new circuit we discovered for task-specific fine-tuning at Epoch 3. The emerging, marked in blue, circuit components formed performed similar mechanisms as the prior circuit components.

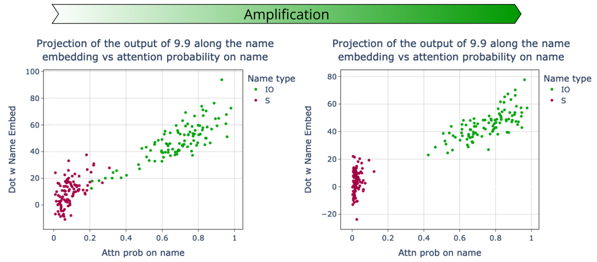
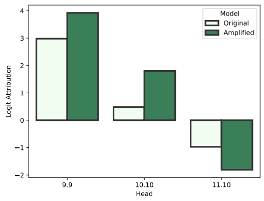


Figure 4. Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9



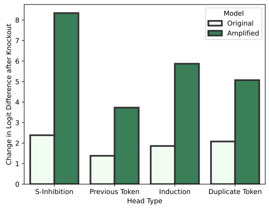


Figure 5. Change in Logit Attribution of heads L9H9, L11H10, L10H10 in the original and amplified model

Figure 6. Change in absolute Logit Difference in the original model vs amplified model after ablating groups of heads

havior in the case of Negative Name Mover Heads, see Appendix H. This implies that this head writes more strongly to the residual stream as the direct logit attribution[1] 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 5. 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 Appendix H and Figure 5. We now see that the attention probability w.r.t to 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 attribution, from 0.4 to 1.8 on the IOI task. Furthermore, another characteristic of Name Mover Heads is that their exists behaviours of Self-Repair when ablated (Wang et al., 2022), we see a similar characteristic for the case L10H10 when amplified, hence we claim that another mechanism of circuit amplification is the change in the mechanism of some of the backup components to that of the component they are backing up. We find evidence of *amplification* across various model mechanisms, which we test by ablating groups of heads in the original model and the fine-tuned models and measuring the change in the logit difference in the circuits performance. As the number of model components performing the task increases, for fair comparison, we only consider the heads in the orig-

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

inal model for each group, see Figure 6. These findings generalize across epochs.

5. Circuit Poisoning

Given the knowledge of circuit amplification, we aim to fine-tune the model with various corrupted augmentations of the IOI task and utilize Path Patching and Activation Patching 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 augmentations. We find that in some augmentations the model behavior is not corrupted and in other augmentations, the corruption can be traced back to changes in the original circuit. Furthermore, we analyze the effect of corruption across various epochs, analyze the underlying mechanism of change across time, and discuss our findings in the following subsections.

Data Corruption: Name Moving Behavior. As anticipated, after fine-tuning, this corrupted dataset effectively suppresses the output of the IO token. Notably, in the case of 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. However, this capability completely degrades over time. To elucidate the underlying mechanisms, we present a detailed case analysis of the fine-tuning process with 3 epochs on the corrupted data-augmented dataset, revealing insightful changes in the model's internal workings, in this section.

Our investigation reveals a crucial insight: the model does **not introduce** novel mechanisms to mitigate performance degradation on the task. Instead, it relies on diminishing the capabilities of specific attention heads that underlie a task-related mechanism and altering the functionality of pre-existing circuit components. Notably, the most affected components are the Name Mover Heads and Negative Name Mover Heads, which completely lose their ability to copy the IO token (see Appendix I and Figure 7). We trace the 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 and Negative 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 Appendix I and Figure 8. We find that this mechanism of corruption extends to Backup Name Mover Heads. Now we trace the information flow back from the S-Inhibition Heads to understand the affect of corruption on the prior heads and find that the functionality of the Induction Heads, Previous Token Heads and remain the same, hence we ask the question: *What is affecting the queries of the S-Inhibition Heads?*. To answer this, employ Path Patching on query vector for the S-Inhibitions and find that Induction Heads, Previous Token Heads and Duplicate Token don't write a strong enough signal to bias the queries of the S-Inhibition Head and hence, S-Inhibition Head attends strongly to both IO and S tokens. This hints that model poisoning, mechanistically, alters/diminish very localized model behaviors that affect the final output, instead of adding novel mechanism to corrupt the model.

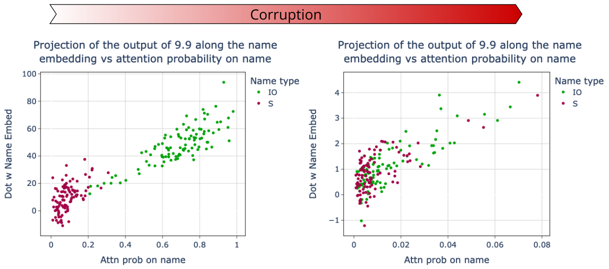


Figure 7. **Name Moving:** Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9

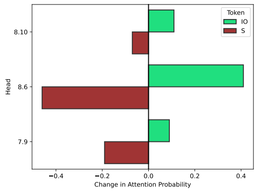


Figure 8. Name Moving: Change in the attention probability of the L8H10 (S-Inhibition) on the IO and S token [*Original - Corrupted*]

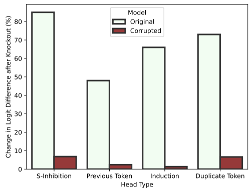


Figure 9. Subject Duplication: Change in Logit Difference after ablating groups of heads.

Data Corruption: Duplication Behavior. In the case of this particular corrupted data augmentation, we find that the model's performance on the prior task is neither enhanced nor corrupted across a variety of epochs. We hypothesize that this is because the particular mechanism of corruption for this task is via corrupting the behavior of the Name Mover Heads. Whereas, this particular data augmentation concerns itself with targeting the Duplicate detection and Inhibition of the S token behavior of the model, which is not affected by corruption. This observation holds even after finetuning for longer epochs.

Data Corruption: Subject Duplication Task. Upon applying this data augmentation strategy and fine-tuning on the corrupted dataset results in a rapid and significant degra-

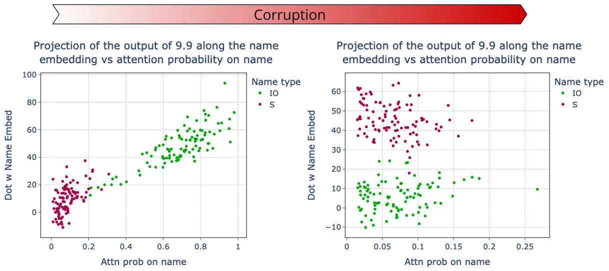


Figure 10. Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9

dation of model performance, characterized by an average logit difference of -11.06 after 5 epochs. Analysis reveals that the Name Mover Heads are most affected, exhibiting a modified attention pattern wherein they attend to both duplicates of the S token and a single instance of the IO token. This altered attention pattern yields a suppressed logit for the IO token and an enhanced logit for the S 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 counterintuitive as these components were originally writing in the direction of the S token, however after finetuning on the corrupted data imputation, these heads now write in the direction of the IO token, see Figure 10 and Appendix I.

Finally, we find that the mechanism of the S-Inhibition heads is almost completely suppressed, even though they still bias the query of the Name Mover Heads and Negative Name Mover Heads, the impact of the bias is very little, when compared to the original circuit. Similar to the previous observation, the mechanism of corruption is very **local** to certain model components, however, unlike the prior case, only the mechanism of the Name Mover Heads, Negative Name Mover Heads and Backup Name Mover heads is changed, while the mechanism of the S-Inhibition Heads (and other heads) is suppressed see Figure 9.

6. Neuroplasticity

After corruption, we study relearning the IOI task via finetuning on the original dataset. We investigate whether the corrupted model can recover its performance and explore changes in mechanisms between the retrieved and original models. Focusing on the two data imputations (excluding Duplication Data Augmentation), we define the fine-tuned model as the *post-reversal* model.

Data Corruption: Name Moving Behavior: 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 (see Appendix G for full circuit details). Taking the case of the Name Mover Heads,**L9H9**, we see that in the *post-reversal* model, a recovery of the original mechanism of the head, moreover, we record *amplification* of the original model's mechanism as previously seen in section 4, see Figure 11. This behavior of neuroplasticity is also recorded in the Backup Name Mover Heads, see Appendix G for the full explanation.

Data Corruption: Subject Duplication Task. The above finding generalizes for the Subject Duplication Corruption as well, with the model retrieving the original mechanisms and furthermore, amplification occurs in this case as well, see Appendix G for further details.

7. Related Work

Fine-Tuning enhances language model performance for specific tasks and general settings (Christiano et al., 2017; Gururangan et al., 2020; Madaan et al., 2022; Touvron et al., 2023). Research has explored its effects on model capabilities, revealing insights into OOD detection (Uppaal et al., 2023; Zhang et al., 2024), domain adaptation via shifting weights to task-specific sub-domains (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 in generalized domains like code, mathematics, and instructions (Prakash et al., 2024) and for synthetic tasks(Jain et al., 2023; Lindner et al., 2024). However, uncertainties remain regarding how fine-tuning enhances mechanisms and if it applies to specific tasks. Our work addresses this by explaining the enhancement and generalizing it to task-specific cases.

Model Poisoning Prior work on corrupting model behaviors utilize meta-learning to poison neural networks (Huang et al., 2020), while other works studied poisoning under token-limits(He et al., 2024), works have also highlighted scaling poisoning method to web-scale training data (Carlini et al., 2023), research has been done in poisoning during instruction tuning (Shu et al., 2023; Wan et al., 2023). Recently, work has been done to make models more susceptible to backdoors via model editing(Li et al., 2024). 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).

Mechanistic Interpretability In addition to reverse-engineering the mechanisms of certain tasks (Wang et al., 2022; Hanna et al., 2024; García-Carrasco et al., 2024; Lindner et al., 2024; Prakash et al., 2024), prior interpretability research, has directed it's attention to mechanistically understanding tasks under phenomenons such as grokking (Nanda

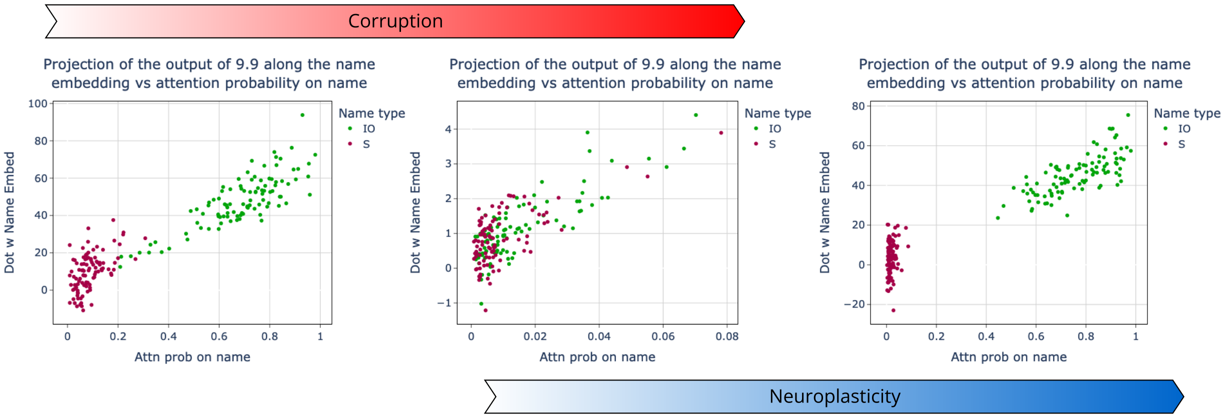


Figure 11. Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L9H9, corruption on **Name Move** augmentation.

et al., 2023; Zhong et al., 2024), while some focuses on exploring specific phenomenons such as Self-Repair(Rushing & Nanda, 2024), 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. Discussion

In this work, we took the case of Indirect Object Identification task on GPT2-small and analyzed the change in its mechanism under task-specific fine tuning, task-specific corruption and ultimately, relearning the task. In our investigation, we record an enhancement of the underlying mechanisms of the model on the task, Circuit Amplification, we quantify and discover the underlying mechanism behind Circuit Amplification and call it *amplification*, which primarily increases the number of components performing similar mechanisms as the original circuit and increase the capabilities of the underlying mechanisms. We record this finding across various epochs. Furthermore, we, with knowledge of the circuit, *a priori*, construct poisonous data augmentations and utilize task-specific fine-tuning on these variations to corrupt model performance on the IOI task, furthermore, we describe the underlying mechanism behind various corrupted augmentations and record the effect of corruption to be localized to circuit components, primarily degrading present components mechanisms instead of creating novel mechanism to counter present mechanisms. Finally, we discover behaviors of neuroplasticity in model mechanisms,i.e, the model quickly relearns the original mechanism after corruption with no change to the underlying mechanisms. Notably, we provide some initial investigations on enhancement of Self-Repair (Rushing & Nanda, 2024) on task-specific fine-tuning,see Appendix D , we believe analyzing the effects of fine-tuning on self-repair can be relevant future work.

Limitations: Our work focuses on a single task on a specific architecture. Significant additional work is needed to scale/replicate 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.

9. Contribution Statement

- Vishnu Kabir Chhabra lead the project in writing, ideas and experimental results, writing most of the paper and was the creative lead for the experiments and research direction. Vishnu also discovered circuit amplification, corruption mechanism and behavior of neuroplasticity in model mechanisms.

- Ding Zhu helped with multiple experiments in regards to minimality of the discovered circuits, tested the circuits discovered for correctness and wrote Appendix A, Appendix B, Appendix C and Appendix F.

- Mohammad Mahdi Khalili was the supervisor of the project providing guidance and direction at every step of the project and helped in formulating the research agenda.

10. Acknowledgement

This material is based upon work 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

Alon, U., Xu, F., He, J., Sengupta, S., Roth, D., and Neubig, G. Neuro-symbolic language modeling with automaton-augmented retrieval. In *International Conference on Machine Learning*, pp. 468–485. PMLR, 2022.

Carlini, N., Jagielski, M., Choquette-Choo, C. A., Paleka, D., Pearce, W., Anderson, H., Terzis, A., Thomas, K., and Tramèr, F. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*, 2023.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Chughtai, B., Chan, L., and Nanda, N. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pp. 6243–6267. PMLR, 2023.

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

Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Davies, X., Nadeau, M., Prakash, N., Shaham, T. R., and Bau, D. Discovering variable binding circuitry with desiderata. *arXiv preprint arXiv:2307.03637*, 2023.

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

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

García-Carrasco, J., Maté, A., and Trujillo, J. C. How does gpt-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *International Conference on Artificial Intelligence and Statistics*, pp. 3322–3330. PMLR, 2024.

Geiger, A., Lu, H., Icard, T., and Potts, C. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

Geiping, J., Fowl, L., Somepalli, G., Goldblum, M., Moeller, M., and Goldstein, T. What doesn't kill you makes you robust (er): How to adversarially train against data poisoning. *arXiv preprint arXiv:2102.13624*, 2021.

Gueta, A., Venezian, E., Raffel, C., Slonim, N., Katz, Y., and Choshen, L. Knowledge is a region in weight space for fine-tuned language models. *arXiv preprint arXiv:2302.04863*, 2023.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

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

He, J., Jiang, W., Hou, G., Fan, W., Zhang, R., and Li, H. Talk too much: Poisoning large language models under token limit. *arXiv preprint arXiv:2404.14795*, 2024.

Huang, W. R., Geiping, J., Fowl, L., Taylor, G., and Goldstein, T. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020.

Jain, S., Kirk, R., Lubana, E. S., Dick, R. P., Tanaka, H., Grefenstette, E., Rocktäschel, T., and Krueger, D. S. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *arXiv preprint arXiv:2311.12786*, 2023.

Karapantelakis, A., Alizadeh, P., Alabassi, A., Dey, K., and Nikou, A. Generative ai in mobile networks: a survey. *Annals of Telecommunications*, 79(1):15–33, 2024.

Li, Y., Li, T., Chen, K., Zhang, J., Liu, S., Wang, W., Zhang, T., and Liu, Y. Badedit: Backdooring large language models by model editing. *arXiv preprint arXiv:2403.13355*, 2024.

Lindner, D., Kramár, J., Farquhar, S., Rahtz, M., McGrath, T., and Mikulik, V. Tracr: Compiled transformers as a laboratory for interpretability. *Advances in Neural Information Processing Systems*, 36, 2024.

Liu, X., Xu, P., Wu, J., Yuan, J., Yang, Y., Zhou, Y., Liu, F., Guan, T., Wang, H., Yu, T., et al. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*, 2024.

Lo, M., Cohen, S. B., and Barez, F. Large language models relearn removed concepts. *arXiv preprint arXiv:2401.01814*, 2024.

Madaan, A., Zhou, S., Alon, U., Yang, Y., and Neubig, G. Language models of code are few-shot commonsense learners. *arXiv preprint arXiv:2210.07128*, 2022.

McDougall, C., Conmy, A., Rushing, C., McGrath, T., and Nanda, N. Copy suppression: Comprehensively understanding an attention head. *arXiv preprint arXiv:2310.04625*, 2023.

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

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Merullo, J., Eickhoff, C., and Pavlick, E. Circuit component reuse across tasks in transformer language models. *arXiv preprint arXiv:2310.08744*, 2023.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Olah, C. Mechanistic interpretability, variables, and the importance of interpretable bases, 2022. URL `https://transformer-circuits.pub/2022/mech-interp-essay/index.html`.

OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2023.

Prakash, N., Shaham, T. R., Haklay, T., Belinkov, Y., and Bau, D. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *arXiv preprint arXiv:2402.14811*, 2024.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., and Azam, S. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 2024.

Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., and Nanda, N. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024.

Rushing, C. and Nanda, N. Explorations of self-repair in language models. *arXiv preprint arXiv:2402.15390*, 2024.

Shu, M., Wang, J., Zhu, C., Geiping, J., Xiao, C., and Goldstein, T. On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems*, 36: 61836–61856, 2023.

Sun, X., Li, X., Meng, Y., Ao, X., Lyu, L., Li, J., and Zhang, T. Defending against backdoor attacks in natural language generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5257–5265, 2023.

Syed, A., Rager, C., and Conmy, A. Attribution patching outperforms automated circuit discovery. *arXiv preprint arXiv:2310.10348*, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wan, A., Wallace, E., Shen, S., and Klein, D. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, pp. 35413–35425. PMLR, 2023.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Yan, L., Zhang, Z., Tao, G., Zhang, K., Chen, X., Shen, G., and Zhang, X. Parafuzz: An interpretability-driven technique for detecting poisoned samples in nlp. *Advances in Neural Information Processing Systems*, 36, 2024.

Yang, H., Zhang, Y., Xu, J., Lu, H., Heng, P. A., and Lam, W. Unveiling the generalization power of fine-tuned large language models. *arXiv preprint arXiv:2403.09162*, 2024.

Zhang, A., Xiao, T. Z., Liu, W., Bamler, R., and Wischik, D. Your finetuned large language model is already a powerful out-of-distribution detector. *arXiv preprint arXiv:2404.08679*, 2024.

Zhao, S., Gan, L., Tuan, L. A., Fu, J., Lyu, L., Jia, M., and Wen, J. Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.12168*, 2024.

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

Zhou, P., Wang, L., Liu, Z., Hao, Y., Hui, P., Tarkoma, S., and Kangasharju, J. A survey on generative ai and llm for video generation, understanding, and streaming. *arXiv preprint arXiv:2404.16038*, 2024.

Zhu, B., Qin, Y., Cui, G., Chen, Y., Zhao, W., Fu, C., Deng, Y., Liu, Z., Wang, J., Wu, W., et al. Moderate-fitting as a natural backdoor defender for pre-trained language models. *Advances in Neural Information Processing Systems*, 35:1086–1099, 2022.

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 hyper-parameter 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 batch-size = 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.

C. Path Patching and Knockout

Path patching is a method to search the attention head h which directly affect the model's logits. 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. 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 & 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.

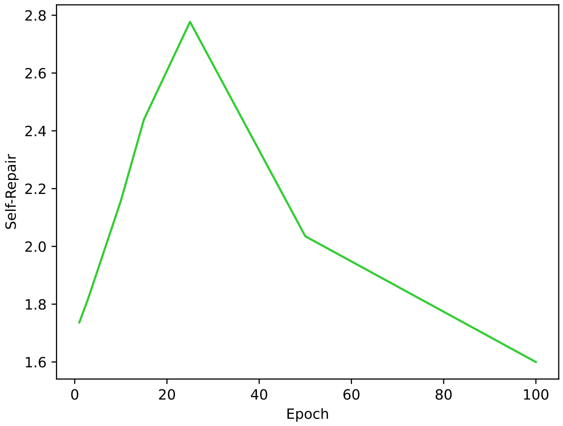


Figure 12. Self-Repair Enhancement over Time for L9H9

We find that capacity of self-repair increases linearly 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 & 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).

Table 2. The accuracy of the model, the circuit, faithfulness, and sparsity of the circuit discovered on various datasets/methods of fine-tuning.

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%

F. Circuit Evaluation

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

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.

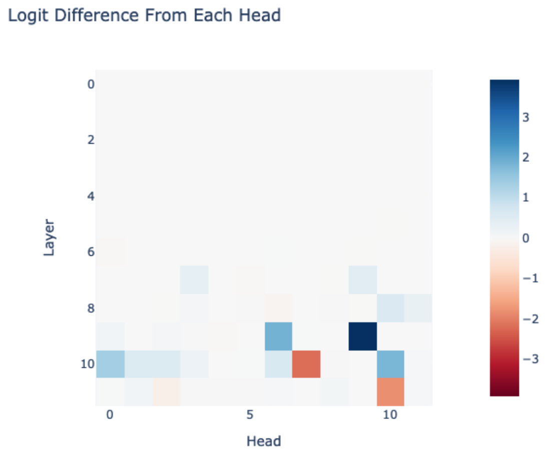


Figure 13. : Isolating Heads with highest direct logit contribution to the task: Name Mover Heads and Negative Name Mover Heads

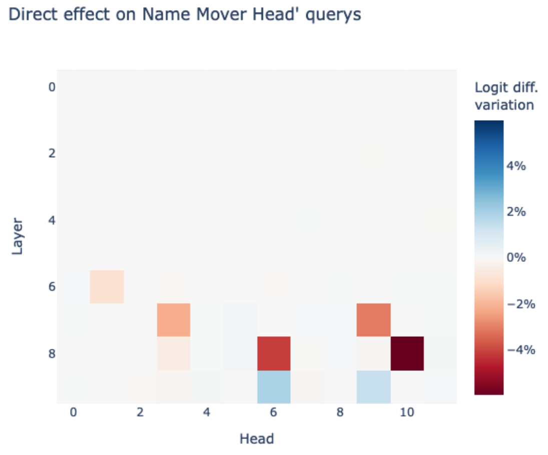


Figure 14. : Isolate important heads that most impact the queries of Name Mover Heads: S-Inhibition Head

We initially, analyze the attention patterns of the heads that have the highest logit attribution to the task. 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

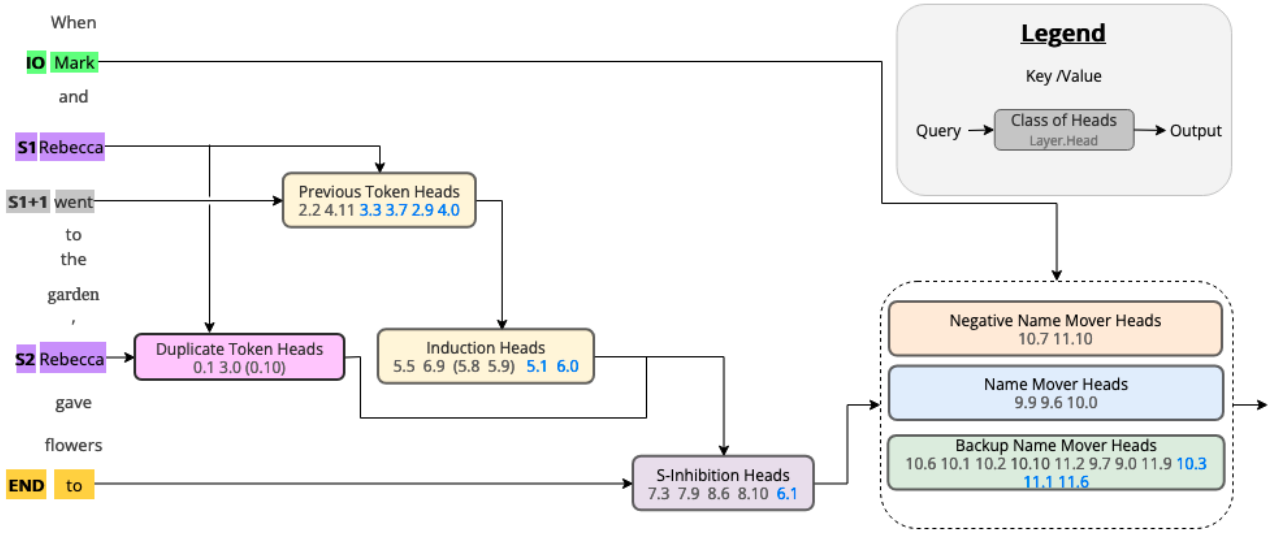


Figure 17. The circuit discovered *post-reversal* after corruption on Name Moving Augmentation, the new components are marked in blue.

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. 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:

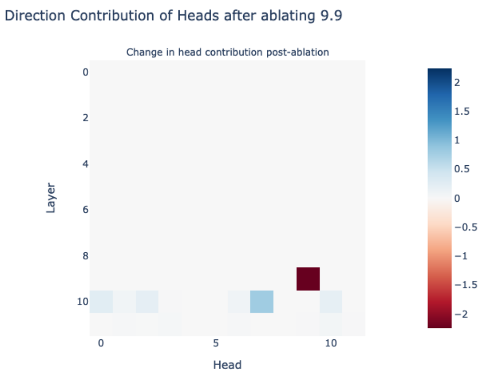


Figure 15. Discovering Backup Name Mover Heads

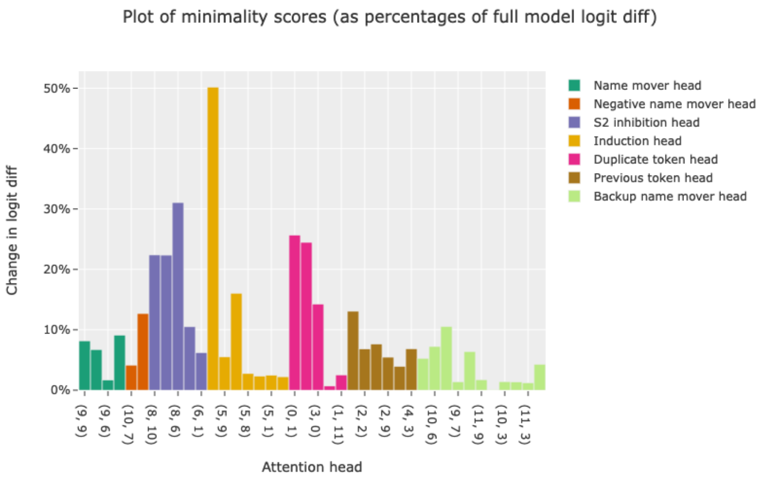


Figure 16. Minimality Scores for the discovered circuit.

Neuroplasticity:

Data Augmentation: Name Moving: We present the circuit for the relearned mechanisms, in the *post-reversal* model. 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

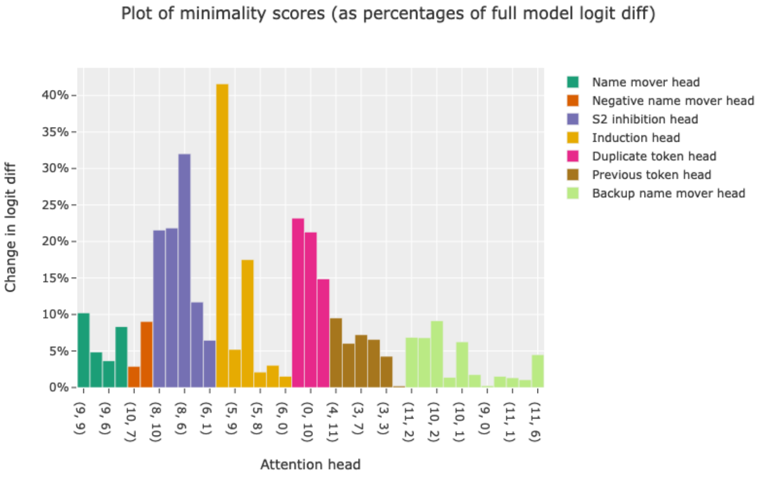


Figure 18. Minimality Scores of the circuit discovered after corruption on Subject Duplication Task.

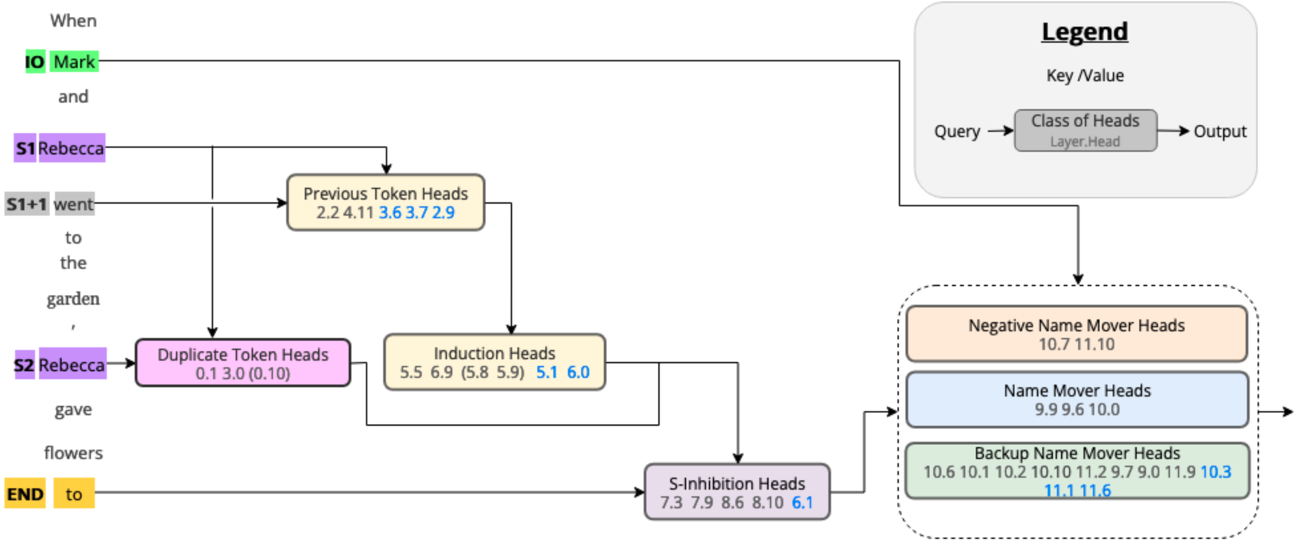


Figure 19. The circuit discovered *post-reversal* after corruption on Subject Duplication Augmentation, the new components are marked in blue.

The faithfulness score of this model is 96% with identical minimality scores as *post-reversal* with Name Moving Behavior.

H. Circuit Amplification

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

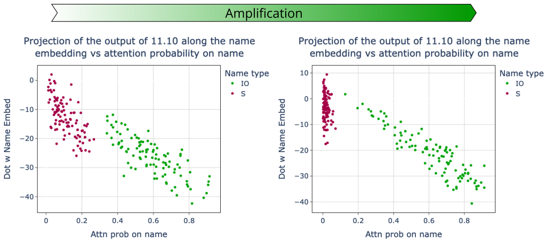


Figure 20. : Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L11H10

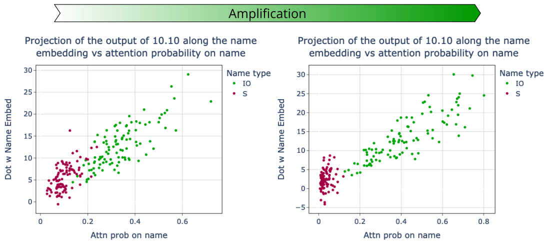


Figure 21. :Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L10H10

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.

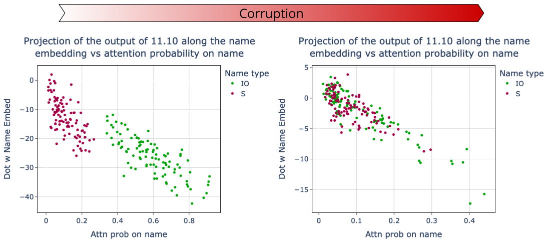


Figure 22. :Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L11H10

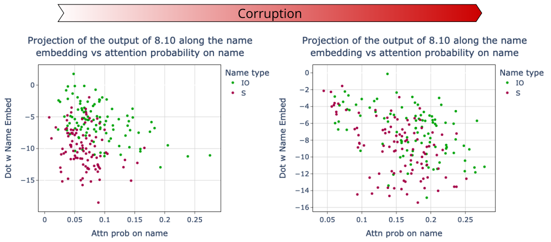


Figure 23. Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L8H10

Subject Duplication Behavior: We now report the degradation of the mechanism of the Negative Name Mover Heads on this task.

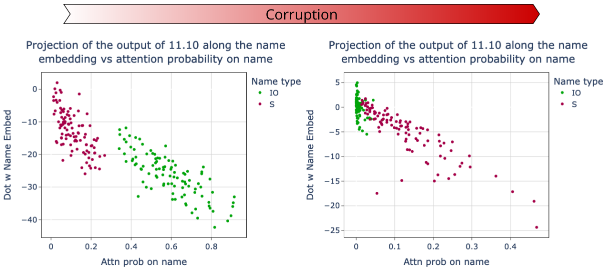


Figure 24. Attention Probability vs Projection of head output along $W_U[IO]$ and $W_U[S]$ for head L11H10