Knowledge Circuits in Pretrained Transformers

Yunzhi Yao[1] **Ningyu Zhang**[1]* **Zekun Xi**[1] **Mengru Wang**[1]
Ziwen Xu[1] **Shumin Deng**[2] **Huajun Chen**[1,3]*
[1] Zhejiang University [2] National University of Singapore, NUS-NCS Joint Lab, Singapore
[3] Zhejiang Key Laboratory of Big Data Intelligent Computing
{yyztodd,zhangningyu}@zju.edu.cn

Abstract

The remarkable capabilities of modern large language models are rooted in their vast repositories of knowledge encoded within their parameters, enabling them to perceive the world and engage in reasoning. The inner workings of how these models store knowledge have long been a subject of intense interest and investigation among researchers. To date, most studies have concentrated on isolated components within these models, such as the Multilayer Perceptrons and attention head. In this paper, we delve into the computation graph of the language model to uncover the knowledge circuits that are instrumental in articulating specific knowledge. The experiments, conducted with GPT2 and TinyLLAMA, have allowed us to observe how certain information heads, relation heads, and Multilayer Perceptrons collaboratively encode knowledge within the model. Moreover, we evaluate the impact of current knowledge editing techniques on these knowledge circuits, providing deeper insights into the functioning and constraints of these editing methodologies. Finally, we utilize knowledge circuits to analyze and interpret language model behaviors such as hallucinations and in-context learning. We believe the knowledge circuits hold potential for advancing our understanding of Transformers and guiding the improved design of knowledge editing[1].

1 Introduction

"Knowledge is power, and when embodied in the form of new technical inventions and mechanical discoveries it is the force that drives history." [1, 2], Bacon's words are vividly re-enacted in the era of Large Language Models (LLMs) [3, 4], as we witness their immense power in reshaping human society and redefining our understanding of machine intelligence. One thing that cannot be denied is that knowledge encapsulated within these models empowers their capabilities in reasoning, perceiving the world, and engaging in human-like communication. Nevertheless, these powerful models are not without their flaws. They still struggle with issues such as hallucinations [5–7], unsafe norms [8, 9], and offensive behaviors [10, 11] and these problems are exacerbated by the enigmatic internal mechanisms of knowledge storage within language models.

Recently, the research community has devoted significant efforts to unraveling the knowledge storage mechanisms of these models. Various studies [12–19] have been conducted to shed light on this intricate process, aiming to enhance our understanding and improve the safety and reliability of language models. The main finding in previous work is that knowledge may primarily stored in the Multilayer Perceptrons (MLPs) of Transformer-based language models. These MLPs function as a key-value neural memory, with knowledge being stored in what are termed "knowledge neurons" (KN). Based on these findings, researchers conduct Knowledge Editing [18, 20] to update the language models' inaccurate facts, bias and unsafe content in their parametric space. Despite the

[*] Corresponding Author.

[1] Code and data are available in https://github.com/zjunlp/KnowledgeCircuits.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

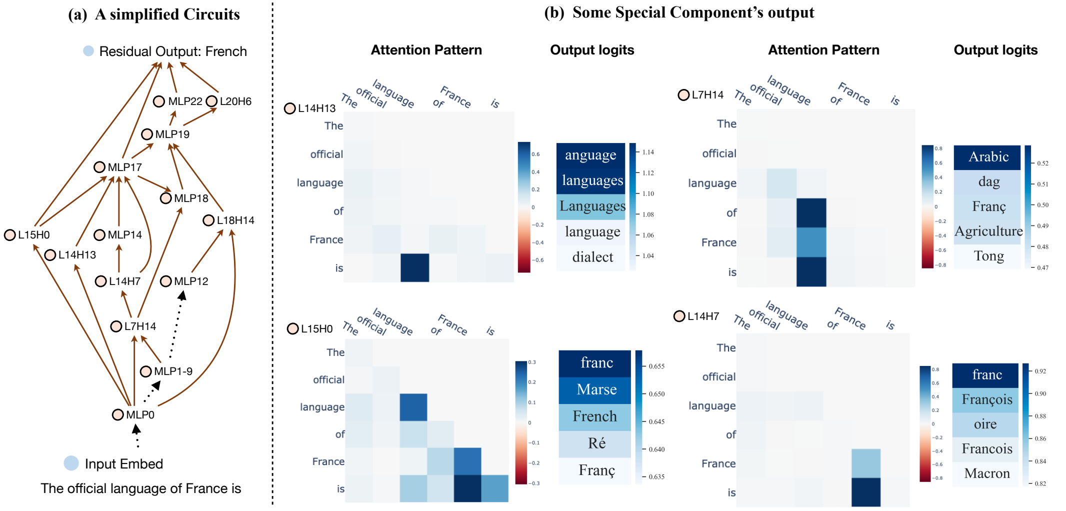


Figure 1: Knowledge circuit obtained from *"The official language of France is French"* in GPT2-Medium. Left: a simplified circuit and the whole circuit is in Figure 9 in Appendix. We use ⇢ to skip some complex connections between nodes. Here, L15H0 means the first attention head in the 15th layer and MLP12 means the multi-perception layer in the 13th layer. Right: the behavior of several special heads. The matrix on the left is the attention pattern of each attention head and the right heapmap demonstrates the output logits of the hean by mapping to the vocabulary space.

initial success of these methods, there are still limitations, such as poor generalization, severe side effects, and failure to effectively utilize edited knowledge [20, 21], which motivate us to re-think previous approaches for interpreting knowledge storage in language models. Note that previous works treat the knowledge blocks as isolated components following the Restorative Theory [22], often focusing on identifying the specific blocks that store particular knowledge. Several works [23, 24] have proposed that different types of knowledge are often located in the same areas, suggesting that the current KN thesis may be an oversimplification.

To this end, instead of solely pinpointing tiny regions where the knowledge expressed can be localized, we aim to explore the cooperation between different components in Transformers like attention heads, MLPs, and embeddings, to understand how the language model stores and expresses the knowledge. Here, we introduce a new perspective: **Knowledge Circuits**, a critical subgraph in the language model to view the knowledge mechanism of Transformers. Note that Circuit, as a subgraph in the computation graph, has gained ever-growing attention in the mechanistic interpretability field [25]. Previous work [26, 27] has found several important circuits for specific tasks like Indirect Object Identification and Color Object Identification. These tasks necessitate the model to search the preceding context for a matching token and copy it into the next token prediction. In this work, we aim to construct knowledge circuits that require the model to utilize stored knowledge for making predictions. Our goal is to better unveil implicit neural knowledge representations, elucidate the internal mechanisms for knowledge editing, and interpret more complex behaviors of language models. Specifically, we leverage factual recall tasks and conduct experiments across various domains, including factual, social bias, linguistic, and commonsense knowledge. We utilize GPT-2 [28] and TinyLLAMA [29] to explore the potential knowledge representations and utilization mechanisms in these models. As shown in Figure 1 (a), we construct knowledge circuits associated with various expressions of knowledge using the existing knowledge stored in the language model. Through those discovered knowledge circuits, we find many interesting phenomena and conclusions as follows:

Knowledge circuits unveil implicit neural knowledge representations. We find that even when the knowledge circuits are used independently, the language model can recall related knowledge with a significant portion of its overall performance, demonstrating the effectiveness of those discovered knowledge representations (circuits). We also delve into specific pieces of knowledge and analyze the information flow within their respective circuits, indicating that the language model tends to aggregate knowledge in the earlier to middle layers and further enhances this information in the later layers. We further uncover several special components (e.g., mover heads and relation heads) in transferring information to the final token position and capturing relational information from the context (Figure 1 (b)).

Knowledge circuits elucidate internal mechanisms for knowledge editing. We conduct experiments to evaluate the impact of current knowledge editing methods on the language models' original knowledge circuits. Empirically, we observe that ROME [18] tends to incorporate edited information primarily at the edited layer. Subsequent mover heads (Appendix B.2) then transport this information to the residual stream of the last token. Conversely, during fine-tuning, the edited token is directly integrated into the language model, exerting a dominant influence on subsequent predictions.

Knowledge circuits facilitate interpreting language model behaviors. We further utilize the knowledge circuits to interpret language model behaviors, such as **hallucination** and **in-context learning**. We observe that when hallucination occurs, the language model fails to correctly transfer knowledge to the final token in the earlier layers. This is evident as the knowledge circuit lacks an effective "mover" head, or the mover head selects incorrect information. Additionally, we notice that several new attention heads emerge in the knowledge circuit during in-context learning.

2 Background: Circuit Theory

2.1 Preliminaries

In the context of neural network interpretability, a circuit can be conceptualized as a human-interpretable subgraph that is dedicated to executing specific tasks within a neural network model [30, 26, 31–33]. When we visualize a neural network model as a connected directed acyclic graph (DAG), denoted as $\mathcal{G}$, the individual nodes represent the various components involved in the forward pass, such as neurons, attention heads, and embeddings. The edges symbolize the interactions between these components, including residual connections, attention mechanisms, and projections. A circuit, represented as $\mathcal{C} \subseteq \mathcal{G}$, emerges as a significant subgraph of $\mathcal{G}$ that is responsible for particular behaviors or functionalities. In this paper, we focus on the Transformer decoder architecture to conduct our experiments. The residual stream of Transformers has been demonstrated to be a valuable tool for mechanistic interpretability in recent works [25, 16]. The Transformer architecture typically starts with token embeddings, followed by a sequence of "residual blocks" and concludes with a token unembedding. Each residual block comprises an attention layer and an MLP layer, both of which "read" their input from the residual stream (via a linear projection) and "write" their output back to the residual stream through an additive projection. We can consider an attention head $A_{l,j}$ (the jth attention head in layer l) as operating on the residual stream from the previous layer, R_{l-1}. Given that $R_0 = I$ (where I represents the input embeddings), we can reinterpret attention head $A_{l,j}$ as processing the cumulative output of all previous attention heads and MLPs and input embedding, treating each node in the previous layers as separate input arguments. Similarly, an MLP node M_l can be seen as operating on the cumulative output of all previous attention heads and MLPs and input embedding, and the output node O operates on the sum of the input embeddings and the outputs of all attention heads and MLPs. The following equations represent the residual connections in the Transformer model, where R_l is the residual stream at layer l, and Input_l^A and Input_l^M are the inputs to the attention and MLP layers, respectively:

$$R_l = R_{l-1} + \sum_j A_{l,j} + M_l, R_0 = I$$

$$\text{Input}_l^A = I + \sum_{l'<l} \left(M_{l'} + \sum_{j'} A_{l',j'} \right)$$

$$\text{Input}_l^M = I + \sum_{l'<l} M_{i'} + \sum_{l' \leq i} \sum_{j'} A_{l',j'}$$

The computational graph $\mathcal{G}$ of the Transformer represents the interactions between attention heads and MLPs. The nodes in $\mathcal{G}$ encompass the input embedding I, attention heads $A_{l,j}$, MLPs M_l, and the output node O, denoted as $N = \{I, A_{l,j}, M_l, O\}$. The edges in the model represent the connections between these nodes, $E = \{(n_x, n_y), n_x, n_y \in N\}$. A circuit $\mathcal{C}$ is meticulously constructed to govern specific behaviors within the model, comprising a selection of nodes $N_{\mathcal{C}}$ and edges $E_{\mathcal{C}}$ that are critical to the successful execution of the tasks at hand, expressed as $\mathcal{C} =< N_{\mathcal{C}}, E_{\mathcal{C}} >$.

2.2 Circuit Discovery

To identify circuits within a language model, a key approach is to examine the model's casual mediation by systematically altering the model's edges and nodes to observe the effects on performance [32, 34, 35]. The underlying principle is that critical edges or nodes are those whose removal results in a notable decline in the model's predictive capabilities. Since the edges in the model's computational graph represent the dependencies between nodes, we can simulate the absence of a particular node-to-node dependency by ablating an edge in the graph. For example, ablating an edge from $A_{i',j'}$ to $A_{i,j}$ involves replacing the contribution of $A_{i',j'}$ in the input to attention head $A_{i,j}$ with zero (in the case of zero ablation) or with the mean value of head $A_{i',j'}$ (in the case of mean ablation). The process of identifying critical edges or nodes through ablation can be broken down into the following steps: i) Overwrite the value of the edge (n_x, n_y) with a corrupted value (either zero or mean ablation), ii) Perform a forward pass through the model with the altered graph, iii) Compare the output values of the modified model with those of the original model using a chosen metric S (Details in Eq.1). If the performance change is below a predefined threshold τ, we can consider the edge non-critical and remove it to obtain a new subgraph $\mathcal{G}/(n_x, n_y)$. In addition to ablation-based methods, recent works have also explored the use of sparse auto-encoders [36, 37] to identify circuits within language models. This approach involves training an auto-encoder to learn a sparse representation of the model's internal structure, which can help reveal the underlying circuitry responsible for specific behaviors or functionalities.

3 Knowledge Circuits Discovery in Transformers

3.1 Knowledge Circuits Construction

Unlike previous work [12, 18], which managed to find out the specific areas that store knowledge, we pay extra heed to the information flow that activates subsequent knowledge for answering questions. Similar to [38, 26], we write language model as a graph consisting of the input, the output, attention heads, and MLPs by considering a "residual rewrite" of the model's computational structure. For example, this residual rewrite gives us a nearly-dense graph in GPT2-medium: one between every pair of (attention head, MLP, input, and output) nodes, except for attention heads in the same layer, which do not communicate with each other. In our paper, we concentrate on the task of answering factual open-domain questions, where the goal is to predict a target entity o given a subject-relation pair (s, r). A knowledge triplet $k = (s, r, o)$ is often presented to the model in the form of a natural language prompt for next token prediction (e.g., *"The official language of France is ____"*). The model $\mathcal{G}$ is expected to generate the target entity, which is consistent with the language model's pretraining format. To identify the circuit that is critical for predicting the target entity o for a given subject-relation pair (s, r), we ablate each special edge $e_i = (n_x, n_y)$ in the computation graph $\mathcal{G}$. We then measure the impact of ablating the edge (**zero ablation** in our implementation) on the model's performance using the MatchNLL loss [32] for the target o:

$$S(e_i) = \log(\mathcal{G}(o|(s,r))) - \log(\mathcal{G}/e_i(o|(s,r))) \tag{1}$$

If the score $S(e_i)$ is less than the predefined threshold τ, we consider the edge to be non-critical and remove it from the computation graph, updating the temporary circuit $\mathcal{C}_{temp} \leftarrow \mathcal{G}/e_i$. We first sort the graph by topological rank following Conmy et al. [32] and traverse all edges in this manner, We derive a circuit $\mathcal{C}_k$ that contributes to representing the knowledge necessary to answer the factual question:

$$\mathcal{C}_k =< N_k, E_k > \tag{2}$$

Here, $\mathcal{C}_k$ is the circuit for the knowledge triplet k, consisting of the nodes N_k and edges E_k that are essential for predicting the target entity o given the subject-relation pair (s, r).

3.2 Knowledge Circuits Information Analysis

Once we have identified the knowledge circuit, we delve deeper into the specific roles and behaviors of each node and edge within the computation graph. Our goal is to comprehend the processing and contribution of each node n_i to the functionality of the circuit. Drawing on the methodologies of previous studies [16, 39, 40], we begin by applying layer normalization to the output of each node n_i and then map it into the embedding space. This is achieved by multiplying the layer-normalized output by the unembedding matrix ($\mathbf{W}_U$) of the language model: $\mathbf{W}_U \, \text{LN}(n_i)$. This transformation allows us to inspect how each component writes information to the circuit and how it influences subsequent computational steps. By understanding the nodes' behavior in the circuit, we can better comprehend the circuit's structure and the key points where information is aggregated and disseminated.

Table 1: **`Hit@10`** of the Original and Circuit Standalone performance of knowledge circuit in GPT2-Medium. **The result for D_{val} being 1.0 indicates that we select the knowledge for which the model provides the correct answer to build the circuit.**

Type	**Knowledge**	**#Edge**	D_{val}		D_{test}		
			Original($\mathcal{G}$)	Circuit($\mathcal{C}$)	Original($\mathcal{G}$)	Random	Circuit($\mathcal{C}$)
Linguistic	Adj Antonym	573	0.80	1.00 ↑	0.00	0.00	0.40 ↑
	word first letter	432	1.00	0.88	0.36	0.00	0.16
	word last letter	230	1.00	0.72	0.76	0.00	0.76
Commonsense	object superclass	102	1.00	0.68	0.64	0.00	0.52
	fruit inside color	433	1.00	0.20	0.93	0.00	0.13
	work location	422	1.00	0.70	0.10	0.00	0.10
Factual	Capital City	451	1.00	1.00	0.00	0.00	0.00
	Landmark country	278	1.00	0.60	0.16	0.00	0.36 ↑
	Country Language	329	1.00	1.00	0.16	0.00	0.75 ↑
	Person Native Language	92	1.00	0.76	0.50	0.00	0.76 ↑
Bias	name religion	423	1.00	0.50	0.42	0.00	0.42
	occupation age	413	1.00	1.00	1.00	0.00	1.00
	occupation gender	226	1.00	0.66	1.00	0.00	0.66
	name birthplace	276	1.00	0.57	0.07	0.00	0.57 ↑
Avg			0.98	0.73	0.44	0.00	0.47 ↑

3.3 Knowledge Circuits Experimental Settings

Implementations. We conduct experiments on GPT-style models, including GPT-2 medium and large. We also conduct primary experiments on TinyLLaMA [29] to validate the effectiveness of different architectures. We utilize the Automated Circuit Discovery [32] toolkit to build a circuit as an initiative of our analysis and leverage transformer lens [41] to further analyze the results. Specifically, we simply employ the MatchNLL [32] as the metric to detect the effect of the given node and edge and use **zero ablation** to knock out the specific computation node in the model's computation graph.

Metrics. A discovered knowledge circuit is deemed an accurate representation of a specific area within the transformer's knowledge storage, thus, it should be capable of representing the knowledge independently. Following [32], we leverage the completeness of a circuit, which refers to its ability to independently reproduce the behavior or predictions of the full model for the relevant tasks. This property is assessed by examining whether the identified subgraph corresponds to the underlying algorithm implemented by the neural network. To evaluate completeness, we first construct the circuit using the validation data D_{val} for a specific knowledge type and then test its performance on the test split D_{test} in isolation. By doing so, we can observe any changes in performance compared to the original model. We use the **`Hit@10`** metric to measure the rank of the target entity o among the top 10 predicted tokens:

$$\texttt{Hit@10} = \frac{1}{|V|}\sum_{i=1}^{|V|} \mathrm{I}\left(\mathrm{rank}_o \leq 10\right) \tag{3}$$

Here, $|V|$ represents vocabulary size, and rank_o is the rank of the target entity o in predictions.

Dataset. In this work, we focus on the **knowledge that is already stored in the language model**. We utilize the dataset provided by LRE [42] and consider different kinds of knowledge, including linguistic, commonsense, fact, and bias. We evaluate whether the knowledge is present in the language model's parameters under zero-shot settings using the **`Hit@10`** metric to sample knowledge from the validation set, which is used to construct the knowledge circuit. The data statistics are in Appendix A.

4 Knowledge Circuits Unveil Implicit Neural Knowledge Representations

Knowledge Circuits Evaluation. We report the results of GPT2-Medium in Table 1, which indicates that with only less than 10% of the original knowledge circuit's subgraph, the model can maintain over 70% of its original performance. Additionally, we compute the random circuits by randomly deciding whether the edge should be removed and making sure the graph is connected. The random circuit is the same size as the circuit we discovered using our method. From the table, we can see that the random circuit failed to maintain the model's performance, which further enhanced the robustness and efficacy of our methods. One of the most fascinating observations is the **performance improvement seen on several test datasets**. For instance, the *Landmark-country* relation metric increases from 0.16 to 0.36. This suggests that the discovered knowledge circuits may encapsulate the relevant knowledge, and the model's performance on these tasks could have been hindered by noise from other components. We proceed to analyze the layer distribution of the original model $\mathcal{G}$ to understand the average percentage of nodes that are activated within the circuit for different knowledge domains. From Figure 2, we observe that attention and MLPs are more active in the lower layers of the network, where the language model processes the input and extracts general information. To gain a more comprehensive view of the information processing, we compute the average rank_o change of the target token in the D_{val} across the layers and report the results in Figure 7. This analysis reveals the phenomenon of early decoding [40], suggesting that by the middle to the latest layers, the target entity is already present in the residual stream, and the subsequent layers in the Transformer are designed to increase the probability of the current token (See discussion in the running example).

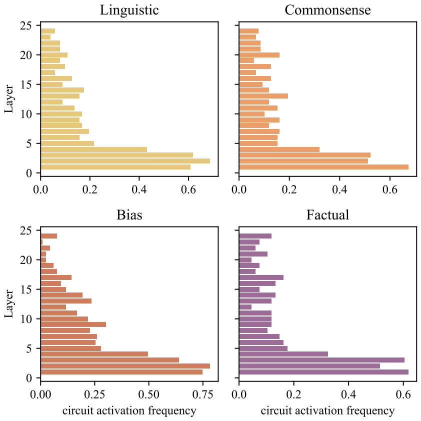


Figure 2: The activated circuit component distributions in Layers in GPT2-Medium.

Special Components in Knowledge Circuits. From the discovered knowledge circuits, we can find several important attention heads that demonstrate specific behavior, including the **mover head** [31], **relation head** [17, 43] and **mixture head** [17, 43] (more definitions in Appendix B.2). Mover Head [31, 27] focuses on the last token of the context and attends to the subject token, functioning as a mover to transfer information, while Relation Head [17] attends to the relation token in the context and produces some relation-related tokens that would guide the behavior of the following components. We think that these components would be accumulated by the MLP in the model, and the behavior of these special heads will be discussed in the running example part. We list some of these special components in Table 4 in Appendix. The different attention heads are responsible for expressing specific types of knowledge and may be activated by different facts. In our experiments with GPT-2 Medium and GPT-2 Large, we find that knowledge is distributed across several layers' attention heads and MLP matrices, suggesting that the target knowledge appears to have been accumulated throughout the GPT-2 model. Conversely, in TinyLLAMA, the special components are more concentrated. As depicted in Figure 7, the rank of the target entity in TinyLLAMA experiences a sharp decline around several layers, whereas in the GPT2 model, the decline is more gradual. We hypothesize that this discrepancy may be attributed to the model's knowledge capacity [44] and warrants further investigation.

A Running Example of Knowledge Circuit. We present a case and analyze the specific behaviors of components within the identified knowledge circuits. In particular, we find some special attention heads in the model such as the mover head and the relation head. We demonstrate the function of these heads in Figure 6 by ablating them from the circuit. Taking the factual knowledge *"The official language of France is French"* as an example, we visualize the knowledge circuit in Figure 1. To express the information flow within the model more effectively, we have plotted the rank and probability of the target entity *o* at each layer when it is mapped into the embedding space, in Figure 3. From this figure, we can see that after MLP 17, the target knowledge emerges as the top token in the residual stream, and after that layer, it undergoes an increased probability. The edges connected to MLP17 are (L14H13 → MLP17), (L14H7 → MLP17), and (L15H0 → MLP17) . Here, the L14H13 is a relation head that focuses on the relation token in the context. The output of this head is relation-related tokens such as *"language"* and *"Language"*. The attention head L14H7 is a mover head that moves the information from the subject position *"France"* to the last token. Previous work [31, 19] has introduced this mover head as an *argument parser*, which moves *"France"* to the last token, and the subsequent MLP conducts a function application to map *"France"* to *"French"*. An intriguing observation is that we can find the output of this head already contains the target answer entity, which significantly contributes to the final output (L14H7 → Output). Also, we see the probability of the subject token in Figure 3 at the last token is nearly zero across these layers. Hence, instead of the *argument parser* function, we consider this mover head as an *extract head* proposed by Geva et al. [13], which aims to extract the related-information from the subject token's position. In the subsequent knowledge editing experiments, we can observe changes in the behavior of these types of heads. Additionally, instead of extraction in the later layers proposed by Geva et al. [13], we notice a gradual decrease in rank across all early-to-middle layers. The MLP17 combines information from previous tokens and integrates this information to prioritize the target token at the top rank.

Interestingly, upon tracing the information flow to L14H7, we discovered that it is predominantly activated by L7H14, a relation head, and its output features several language tokens, such as *"Arabic"*. We hypothesize that L7H14 may function as a signaling mechanism to activate the associated mover head, but this hypothesis necessitates further investigation to be confirmed. After MLP17, several attention heads, such as L18H14 (a relation head) and L20H6 (a mover head), collaborated to further enhance the final prediction of the target entity.

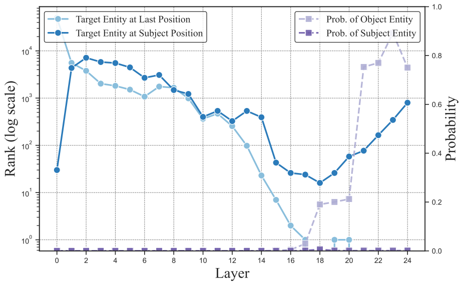


Figure 3: The rank and probability of the target entity o at both the last subject token and the last token position when unembedding the intermediate layer's output for the fact *"The official language of France is French"*.

5 Knowledge Circuits Elucidate Internal Mechanisms for Knowledge Editing

In this section, our objective is to evaluate the impact of previous knowledge editing methods and validate the effectiveness of knowledge circuits. We aim to understand why these methods may fail in certain cases and settings, which can also help deepen the understanding of the knowledge circuit.

Single Factual Knowledge Editing. Here, we adopt the ROME method [18] and FT-M [24], which aim to edit the MLP layers in the language model. The most important hyper-parameter in knowledge editing is the layer, as the same method's performance varies significantly via the layers. Here, we evaluate the performance of different editing layers and their effectiveness. We compare the knowledge circuits computed by the edited model with the original one, and we present results in Figure 4 and report details in Appendix D. As discussed in the previous part, the early-to-middle layers are the main part of aggregating the target entity o to the top rank. In the original model, the probability of the target entity *"Intel"* is nearly zero, and the model fails to elevate it to the top rank in the vocabulary. Editing the model with ROME and FT-M both give us the correct answer but we can view different scenarios for their knowledge circuits. For **ROME**, as the correct information is added to the subject position, we can recognize a **behavior of the Mover Head shifts from copying to extracting the edited information from the subject position**. This information gradually aggregates through the subsequent layers, and by layer 15, *"Intel"* emerges as the top-ranked entity with its probability increasing significantly. Specially, before editing, the mover head L15H3 attends to the *"controller"* token and returns *"controller"* as the output, while in the edited model, the attention head's output moves to the *"Intel"*, which means the model gains the information at the subject space. For FT-M, the edited model tends to directly write the knowledge into the specific component, which would greatly dominate the following component in the model. As shown in Figure 4, the output logits in MLP-0 for *"Intel"* are more than 10, and it emerges as the top rank in the residual stream directly. This phenomenon can be found in different knowledge types and layers and we report results in Appendix D.2. However, the added knowledge may have the risk of influencing unrelated knowledge. When we test another fact *"Windows server"*, the model still tends to give us the *"Intel"* answer, demonstrating the overfitting problem. This finding supports previous analysis regarding the correlation between localization and editing [45], suggesting that edits may not alter the storage but merely add signals into the knowledge circuits.

Multi-hop Factual Knowledge Editing. Multi-hop knowledge editing poses a challenging scenario [20, 21, 46], wherein we edit the model with new knowledge, yet the model struggles to perform reasoning using the edited information. We analyze multi-hop questions in language models [47, 48]

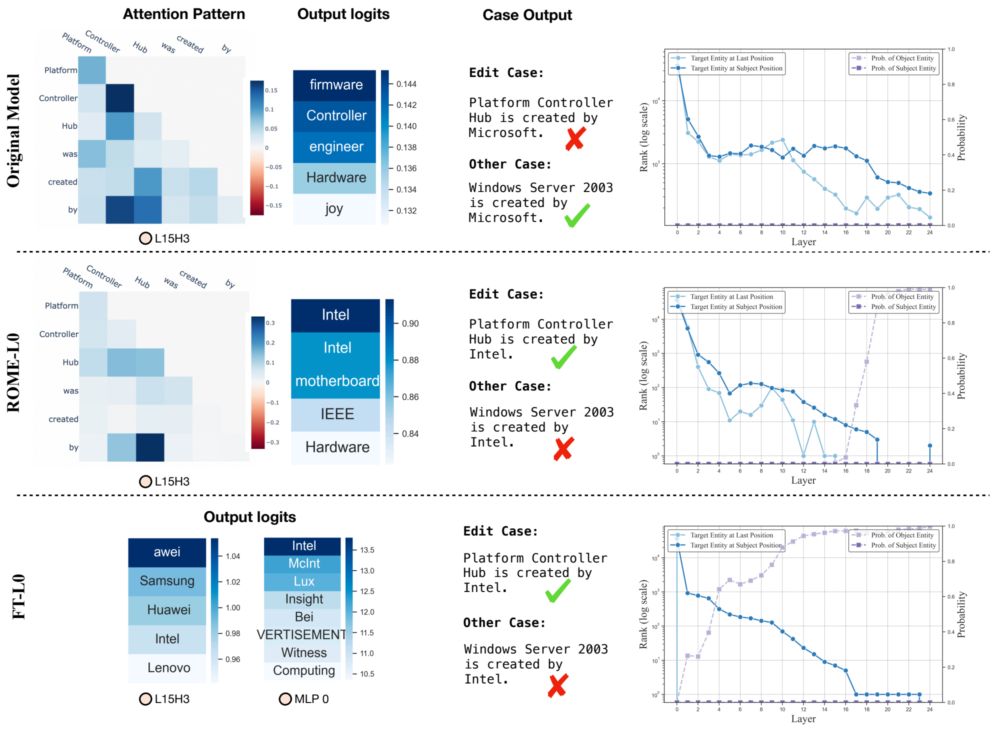


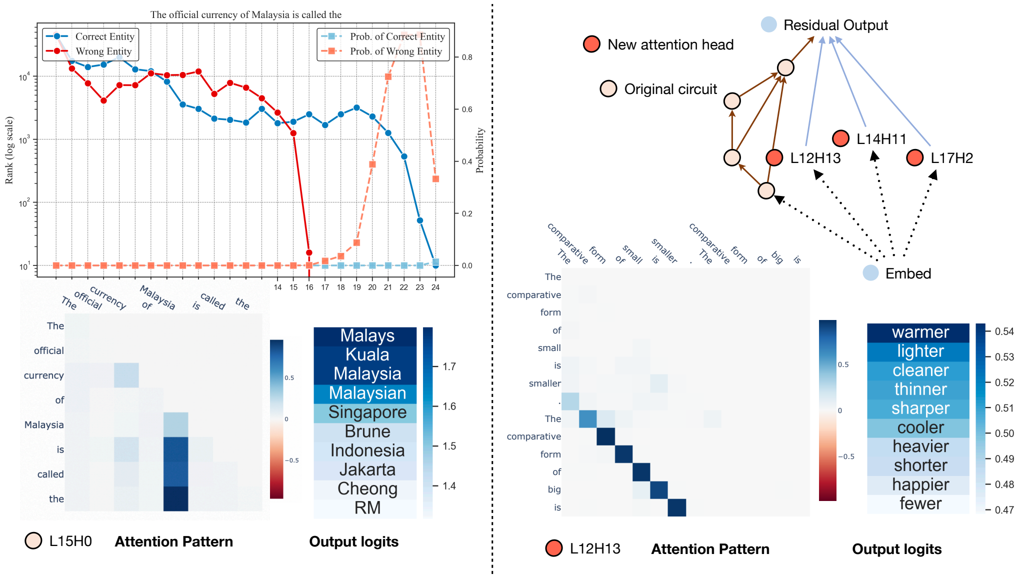
Figure 4: Different behaviors when we edit the language model. In the original model, we can see the mover head L15H3 actually move the original token *"Controller"* and other information, while for ROME, we observe the mover head select the correct information *"Intel"*, which means ROME **successfully added the *"Intel"* to model**. For the FT layer-0 editing, we can find this method **directly write the edited knowledge into edited component**. However, we find these two editing methods would affect other unrelated input *"Windows server is created by?"*

to understand why current editing methods fail in these scenarios. For instance, given the fact (Thierry Mugle, *"home country"*, France), we edit the fact to another country, such as (Thierry Mugle, *"home country"*, France → China). We then assess the model's performance on questions based on the edited knowledge, including *"The official currency of the home country of Thierry Mugle is"* and *"The capital city of the home country of Thierry Mugle is"*. While the unedited model could correctly answer these questions, we observe that the edited model would provide the answer *"China"* for subsequent hop reasoning. We find that the mover head in the original multi-hop reasoning circuit initially extracts the second-hop answer but, after editing, extracts *"China"*, demonstrating that the edited information dominantly saturates and influences the circuit. Furthermore, we observe an intriguing phenomenon: **even in the original model's multi-hop reasoning settings, it would directly provide the answer if we remove the context of the first-hop texts** (Details in Appendix C.1). This further confirms the findings that the model relies on relational and subject-related information, regardless of grammatical adherence.

6 Knowledge Circuits Facilitate Interpreting Language Model Behaviors

In this Section, our aim is to validate whether the identified knowledge circuits are actually utilized by the model when it employs knowledge. To address this, as shown in Figure 5, we investigate three phenomena: hallucination, in-context learning, and reverse relations (Details in Appendix C.3).

Factual Hallucination. If the knowledge is stored and expressed by the circuit we discovered, we aim to discover what happened when the model gave us the incorrect answer. We focus on factual hallucinations, which occur when the model provides an incorrect target entity for a given subject s and relation r. In our experiments (Figure 5 and Appendix C.2), we observe that the model fails to move the correct knowledge to the final token in the earlier layers. This failure is evident as the



(a) A hallucination Case

(b) An in-context learning Case

Figure 5: Left: fact hallucination case *"The official currency of Malaysia is called"*, we observe that, **at layer 15, the Mover Head selects incorrect information**. Right: In-context learning case, we notice that **some new heads focusing on the demonstration appear in the knowledge circuit**.

circuit lacks an effective mover head or the mover head selects incorrect information. For instance, in the prompt *"The official currency of Malaysia is called"*, both the correct answer *"Ringgit"* and the incorrect one *"Malaysian"* are accumulated before layer 15. However, at layer 16, the mover head L15H10 extracts the erroneous information. Despite a rank drop of the true one in layers 20–22, this is insufficient to correct the previous mistake.

In-Context Learning. Despite storing a vast amount of knowledge, a language model may still provide incorrect answers. However, with demonstrations or examples (based on RAG [49]), it can quickly generate correct responses. To this end, we focus on the scenario where the model initially provides an incorrect answer but can then produce the correct response upon receiving the appropriate demonstration. We consider the original knowledge circuit and introduce a new knowledge circuit based on the demonstration. Our analysis reveals that, compared to the zero-shot knowledge circuit, several new attention heads appear in the computation graph when the demonstration is incorporated. We show the behavior of these attention heads in Figure 5. We can see these heads mainly focus on the demonstration's context: *"The comparative form of small is smaller"* and works as the Induction Head [50] that look back over the sequence for previous instances of the current token and find the token that came after it last time. To better view the function of these heads, we conduct experiments by ablating the newly appeared attention head in the ICL circuit in Table 2. We find that compared to the randomly selected attention head by ablating this attention, the probability drops significantly in the prediction, demonstrating the importance of these identified attention heads. These aligned with previous work where Todd et al. [51] have identified a concept known as the Function Vector, which represents the average of some key attention heads and provides the task learning ability.

7 Related Work

Knowledge Mechanism of Transformers. How the language model stores and utilizes knowledge is an ongoing research topic. Previous works find that the MLP in Transformers works as a key-value memory and stores enormous knowledge [12, 15, 14, 18]. As to the relation between entities, Hernandez et al. [42] observes that facts can be decoded linearly from the enriched residual stream of the subject by mapping the subject entity to the object entity. Instead of viewing the knowledge storage in isolation, Geva et al. [13], Lv et al. [31], Yu and Ananiadou [16] find the knowledge is accumulated during the layers. Regarding knowledge analysis, Bayazit et al. [52] also attempts to discover critical knowledge in language models. However, they only consider several layers in the

Table 2: Performance change via ablating the newly appeared attention heads in the ICL circuit and random heads.

	Knowledge	Origin Model	Ablating extra head	Ablating random head
Linguistic	adj_comparative	62.24	32.55	58.18
Commonsense	word_sentiment	89.02	55.50	88.61
	substance_phase	78.74	52.85	71.24
Bias	occupation_gender	86.97	59.54	86.54
Factual	person_occupation	35.17	23.27	31.60

model and use the pruning method, which may overlook the connections between components. More related works can be found in Appendix E.1.

Manipulate Language Models. Recently, many works aim to manipulate the language models to make the model aligned with world knowledge or social value norms, such as knowledge editing [20, 24], machine unlearning [53, 54] and detoxification [55, 56]. Most of these works are elicited by previous knowledge mechanism findings such as knowledge neuron [57]. They modify the MLP in the LLM [18, 12] to change the model's behavior based on specific factual knowledge. However, recent works [58, 59] demonstrate the pivotal role of the attention part in knowledge representation. Hase et al. [45] also observe that the performance of editing within a layer may not reliably pinpoint the location of the fact. In this paper, we try to manipulate specific knowledge of language models via knowledge circuits, including both MLP and attention components across different layers.

8 Conclusion

In this paper, we present a new perspective on knowledge storage based on circuit theory and conduct a preliminary analysis to demonstrate its effectiveness. We found that knowledge circuits in the model are not only responsible for expressing knowledge but can also guide behavior in different settings. We hope these findings can advance our understanding of the knowledge mechanisms of language models and provide insights for better designing and editing language models, enhancing knowledge, and improving reasoning to enhance factuality and alleviate hallucinations.

Limitations and Broader Impacts

In this work, we employ the causal mediation method to automatically construct circuits tailored to specific knowledge domains. However, this circuit discovery-based patching approach is time-intensive. Contemporary research efforts have introduced more efficient methodologies for modeling information flow [60–62]. Additionally, alternative techniques for discovering circuits through mask training [63, 64] and Sparse Auto-Encoders [65, 36] have been proposed, highlighting diverse facets of circuit behavior within large language models (LLMs). We posit that the field of knowledge circuit discovery holds significant potential for advancement. Furthermore, recent studies [66] have developed 'circuit breakers' to manage representations associated with potentially harmful outputs. We hope that our approach can contribute to ensuring the safety and privacy of information, thereby fostering the development of trustworthy AI.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62206246, No. NSFCU23B2055, No. NSFCU19B2027), the Fundamental Research Funds for the Central Universities (226-2023-00138), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Yongjiang Talent Introduction Programme (2021A-156-G), CIPSC-SMP-Zhipu Large Model Cross-Disciplinary Fund, Ningbo Science and Technology Special Projects under Grant No. 2023Z212, Information Technology Center and State Key Lab of CAD&CG, Zhejiang University, NUS-NCS Joint Laboratory (A-0008542-00-00), and the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (FY2023) (Grant A-8001996-00-00). We gratefully acknowledge the support of Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

[1] Francis bacon. https://iep.utm.edu/francis-bacon/.

[2] Francis Bacon. The advancement of learning [1605]. In *Primer of intellectual freedom*, pages 172–192. Harvard University Press, 1949.

[3] OpenAI and the Co-authors. Gpt-4 technical report, 2024.

[4] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *CoRR*, abs/2303.18223, 2023. doi: 10.48550/ARXIV.2303.18223. URL https://doi.org/10.48550/arXiv.2303.18223.

[5] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023.

[6] Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity, 2023.

[7] Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Unified hallucination detection for multimodal large language models. *CoRR*, abs/2402.03190, 2024. doi: 10.48550/ARXIV.2402.03190. URL https://doi.org/10.48550/arXiv.2402.03190.

[8] Helena Bonaldi, Yi-Ling Chung, Gavin Abercrombie, and Marco Guerini. Nlp for counterspeech against hate: A survey and how-to guide, 2024.

[9] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, and Yue Zhao. Trustllm: Trustworthiness in large language models. *CoRR*, abs/2401.05561, 2024. doi: 10.48550/ARXIV.2401.05561. URL https://doi.org/10.48550/arXiv.2401.05561.

[10] Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. Safetybench: Evaluating the safety of large language models with multiple choice questions. *CoRR*, abs/2309.07045, 2023. doi: 10.48550/ARXIV.2309.07045. URL https://doi.org/10.48550/arXiv.2309.07045.

[11] Aiqi Jiang and Arkaitz Zubiaga. Cross-lingual offensive language detection: A systematic review of datasets, transfer approaches and challenges, 2024.

[12] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581.

[13] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751.

[14] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3.

[15] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446.

[16] Zeping Yu and Sophia Ananiadou. Locating factual knowledge in large language models: Exploring the residual stream and analyzing subvalues in vocabulary space, 2024.

[17] Bilal Chughtai, Alan Cooney, and Neel Nanda. Summing up the facts: Additive mechanisms behind factual recall in llms, 2024.

[18] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

[19] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple word2vec-style vector arithmetic, 2024.

[20] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.632.

[21] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the Ripple Effects of Knowledge Editing in Language Models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 04 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00644. URL `https://doi.org/10.1162/tacl_a_00644`.

[22] Belinda Hopkins. *Restorative theory in practice: Insights into what works and why*. Jessica Kingsley Publishers, 2015.

[23] Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. What does the knowledge neuron thesis have to do with knowledge? In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=2HJRwwbV3G`.

[24] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024.

[25] 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. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[26] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=NpsVSN6o4ul`.

[27] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=fpoAYV6Wsk`.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[29] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model, 2024.

[30] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

[31] Ang Lv, Kaiyi Zhang, Yuhan Chen, Yulong Wang, Lifeng Liu, Ji-Rong Wen, Jian Xie, and Rui Yan. Interpreting key mechanisms of factual recall in transformer-based language models, 2024.

[32] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf.

[33] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety – a review, 2024.

[34] Judea Pearl. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pages 373–392. 2022.

[35] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

[36] Zhengfu He, Xuyang Ge, Qiong Tang, Tianxiang Sun, Qinyuan Cheng, and Xipeng Qiu. Dictionary learning improves patch-free circuit discovery in mechanistic interpretability: A case study on othello-gpt, 2024.

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

[38] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023.

[39] Shahar Katz and Yonatan Belinkov. VISIT: Visualizing and interpreting the semantic information flow of transformers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14094–14113, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.939. URL https://aclanthology.org/2023.findings-emnlp.939.

[40] nostalgebraist. interpreting GPT: the logit lens. 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[41] Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/neelnanda-io/TransformerLens, 2022.

[42] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *Proceedings of the 2024 International Conference on Learning Representations*, 2024.

[43] Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. A primer on the inner workings of transformer-based language models, 2024.

[44] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. 2024.

[45] Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=EldbUlZtbd.

[46] Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.971.

[47] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning?, 2024.

[48] Tianjie Ju, Yijin Chen, Xinwei Yuan, Zhuosheng Zhang, Wei Du, Yubin Zheng, and Gongshen Liu. Investigating multi-hop factual shortcuts in knowledge editing of large language models, 2024.

[49] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997, 2023. doi: 10.48550/ARXIV.2312.10997. URL `https://doi.org/10.48550/arXiv.2312.10997`.

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

[51] Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. LLMs represent contextual tasks as compact function vectors. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=AwyxtyMwaG`.

[52] Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. Discovering knowledge-critical subnetworks in pretrained language models, 2023.

[53] Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges, 2023.

[54] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12041–12052, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.738. URL `https://aclanthology.org/2023.emnlp-main.738`.

[55] Xinshuo Hu, Dongfang Li, Baotian Hu, Zihao Zheng, Zhenyu Liu, and Min Zhang. Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18252–18260. AAAI Press, 2024. doi: 10.1609/AAAI.V38I16.29784. URL `https://doi.org/10.1609/aaai.v38i16.29784`.

[56] Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. Detoxifying large language models via knowledge editing, 2024.

[57] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics, 2022.

[58] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=aLLuYpn83y`.

[59] Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 342–356, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.26.

[60] Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. *arXiv preprint arXiv:2403.00824*, 2024.

[61] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *NeurIPS Workshop on Attributing Model Behavior at Scale*, 2023. URL `https://openreview.net/forum?id=tiLbFR4bJW`.

[62] Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=TZ0CCGDcuT`.

[63] Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. Finding transformer circuits with edge pruning. *arXiv preprint arXiv:2406.16778*, 2024.

[64] Lei Yu, Jingcheng Niu, Zining Zhu, and Gerald Penn. Functional faithfulness in the wild: Circuit discovery with differentiable computation graph pruning. *arXiv preprint arXiv:2407.03779*, 2024.

[65] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[66] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with short circuiting. *arXiv preprint arXiv:2406.04313*, 2024.

[67] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL `https://aclanthology.org/2023.emnlp-main.298`.

[68] Shiqi Chen, Miao Xiong, Junteng Liu, Zhengxuan Wu, Teng Xiao, Siyang Gao, and Junxian He. In-context sharpness as alerts: An inner representation perspective for hallucination mitigation, 2024.

[69] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Th6NyL07na`.

[70] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layer skip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*, 2024.

[71] Beren and Sid Black. The singular value decompositions of transformer weight matrices are highly interpretable. https://www.lesswrong.com/posts/mkbGjzxD8d8XqKHzA/the-singular-value-decompositions-of-transformer-weight, 2022.

[72] Junlin Zhang. Parametric reflection of the world: Why can gpt generate intelligence through next token prediction. https://zhuanlan.zhihu.com/p/632795115, 2023.

[73] Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GPKTIktA0k.

[74] Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*, 2023.

[75] Maximilian Li, Xander Davies, and Max Nadeau. Circuit breaking: Removing model behaviors with targeted ablation. *Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning*, 2023.

[76] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

[77] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 30–45. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.3. URL https://doi.org/10.18653/v1/2022.emnlp-main.3.

[78] Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17817–17825. AAAI Press, 2024. doi: 10.1609/AAAI.V38I16.29735. URL https://doi.org/10.1609/aaai.v38i16.29735.

[79] Yuheng Chen, Pengfei Cao, Yubo Chen, Yining Wang, Shengping Liu, Kang Liu, and Jun Zhao. The da vinci code of large pre-trained language models: Deciphering degenerate knowledge neurons. *CoRR*, abs/2402.13731, 2024. doi: 10.48550/ARXIV.2402.13731. URL https://doi.org/10.48550/arXiv.2402.13731.

[80] Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=MkbcAHIYgyS.

[81] Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality, 2024.

[82] Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. *CoRR*, abs/2402.18154, 2024. doi: 10.48550/ARXIV.2402.18154. URL https://doi.org/10.48550/arXiv.2402.18154.

[83] Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. *CoRR*, abs/2310.15213, 2023. doi: 10.48550/ARXIV.2310.15213. URL https://doi.org/10.48550/arXiv.2310.15213.

[84] Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *CoRR*, abs/2402.18312, 2024. doi: 10.48550/ARXIV.2402.18312. URL https://doi.org/10.48550/arXiv.2402.18312.

[85] nostalgebraist. interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020.

[86] Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gfFVATffPd.

[87] Fahim Dalvi, Hassan Sajjad, and Nadir Durrani. Neurox library for neuron analysis of deep NLP models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2023, Toronto, Canada, July 10-12, 2023*, pages 226–234. Association for Computational Linguistics, 2023.

[88] Dan Mossing, Steven Bills, Henk Tillman, Tom Dupré la Tour, Nick Cammarata, Leo Gao, Joshua Achiam, Catherine Yeh, Jan Leike, Jeff Wu, and William Saunders. Transformer debugger. https://github.com/openai/transformer-debugger, 2024.

[89] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models, 2024.

[90] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *CoRR*, abs/2209.10652, 2022. doi: 10.48550/ARXIV.2209.10652. URL https://doi.org/10.48550/arXiv.2209.10652.

Appendix

A Implementation Details

Hyper-parameter. The primary hyperparameter for constructing a circuit is the threshold τ used to detect performance drops. Setting τ too high may result in an incomplete circuit, while setting it too low can introduce numerous unnecessary nodes. In our experiment, we test τ values from the set $\{0.02, 0.01, 0.005\}$ to determine the appropriate circuit size for different types of knowledge.

Implementation. We utilize ACDC[2] to construct circuits that encode specific knowledge representations. Additionally, we re-implement the code to compute the relevant dataset and impact metrics for the knowledge used. Since TinyLLAMA[3] incorporates the Grouped Query Attention Mechanism [67], we interleave and repeat the key and value pairs to analyze the specific behavior of each attention head. We use the NVIDIA-A800 (40GB) to conduct our experiments. It took about 1-2 days to compute the circuit for the knowledge type in GPT2-medium.

Dataset Details. All the data used in our paper is sourced from Hernandez et al. [42], with the detailed information provided in Table 6. In the original setting, they use the data for few-shot settings, but in our experiments, we consider zero-shot knowledge storage, so here we sample the data using the `Hit@10` to detect whether the model understands knowledge for the given prompt based on the s and o. We sample the test set in a 1:1 ratio with the validation set to ensure a balanced evaluation.

Category	# Rel.	# Examples	# GPT-2 Corr.
Factual	26	9,696	4,721
Commonsense	8	374	240
Linguistic	6	806	483
Bias	7	213	149

Table 3: Information about the dataset. Table is borrowed from Hernandez et al. [42]

B More Experiment Results

B.1 Rank Change Across Layers

To gain a clearer understanding of the knowledge aggregation phenomenon, we compute the rank of the target entity rank_o within the vocabulary space $|V|$ at the output of each layer. As depicted in Figure 7, we observe that the model initially elevates the target entity to the top ranks of the vocabulary. Once the entity reaches the top of the vocabulary, subsequent layers continue to enhance its probability mass. These findings corroborate previous work by [68–70], who note the substantial difference in logit entropy between layers, contributing to the model's improved prediction for the target entity. In our work, we aim to delve deeper into how the model, as well as specific components within it, give rise to these behaviors.

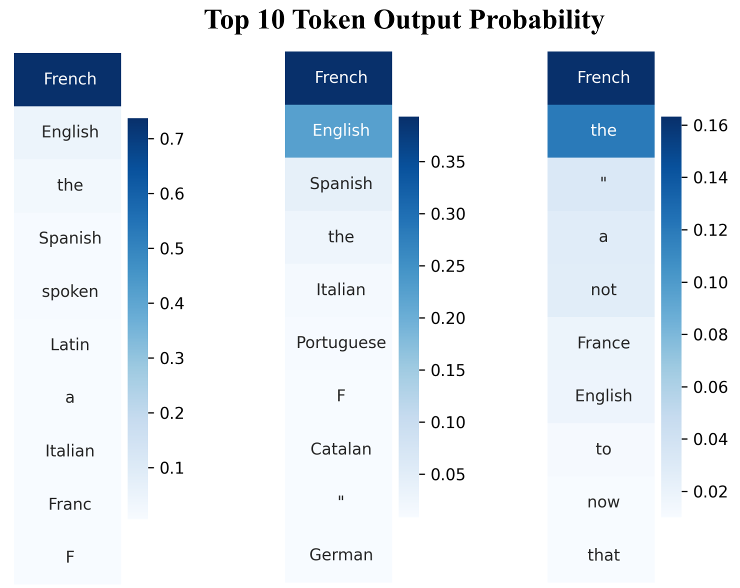
B.2 Special Components in Knowledge Circuit

When zooming into the discovered circuit, we can find several kinds of special attention heads, or MLPs, in the model that play a pivotal role in the final prediction, similar to what previous research has indicated [31]. Apart from mover head and relation head, there is another kind of head named **Mix Head** [17, 43] which would focus on both the relation token and the subject tokens. In our experiments, we found these heads usually work similarly to the mover head. In particular, mover heads contribute more to the subject-specific information, as ablating them would increase other relation-related tokens' probability. Moreover, if we ablate the relation head, we can find the model tends to generate some meaningless tokens instead of the relation information. From Figure 3 6, we can find ablating the mover head would increase the probability of "Italian", "English" and "Spanish", which are not subject-related. While ablating the relation head would lead to the increase of some meaningless words "a", and "that", which are not relation-related.

Lv et al. [31] posits that these mover heads are responsible for extracting the "argument" from the context and passing it for further processing, such as function application. Consequently, Lv et al. [31], Merullo et al. [19] suggests that the MLP within the language model performs a function

[2] `https://github.com/ArthurConmy/Automatic-Circuit-Discovery`

[3] Checkpoint: `https://huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T`



(a) original output (b) ablating mover head (c) ablating relation head

Figure 6: The output of the model. Ablating the mover head would increase the probability of "Italian", "English" and "Spanish", which are not subject-related. While ablating the relation head would lead to the increase of some meaningless words "a", "that", which are not relation-related.

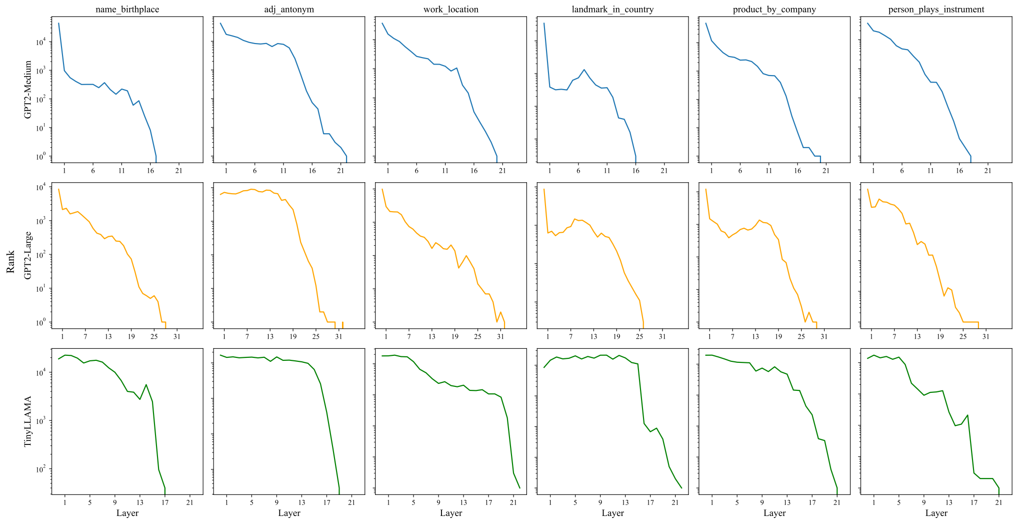
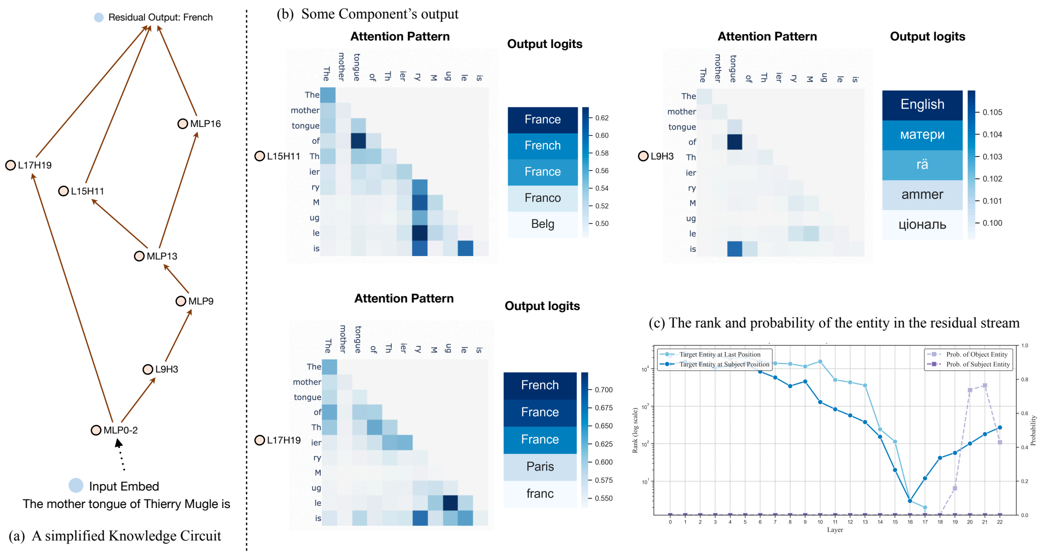


Figure 7: The average rank of the target entity o for the D_{val} in the vocabulary when mapping the output of each layer in the model to the embedding space. We can find that the in GPT2-Medium and GPT2-Large, the model would get the knowledge at middle-to-later layers. While for TinyLLAMA, the layer may be more later.

application that transforms the subject into an object, with the subject's probability ranking higher than the object's. However, these findings are limited to specific cases, as the studies only examine two related tasks, such as capital city identification or color objection. Our analysis suggests that these conclusions may not be universally applicable to all knowledge domains and require further investigation. When we examine the ranks of subject and object entities, we rarely observe an overwhelming superiority at the last token position of the subject token and usually, the probability of the subject token at the last position is uniformly low. In our experiment, we view the model as a collaboration between different nodes in the identified circuit. They perform as different kinds of



(a) A simplified Knowledge Circuit

(b) Some Component's output

(c) The rank and probability of the entity in the residual stream

Figure 8: A simplified knowledge circuit found in TinyLLAMA for the knowledge *"The mother tongue of Thierry Mugler is French"*.

attention heads, like *mover head* and *relation head*. We list some heads that are responsible for the storage of different kinds of knowledge and relations in Table 4.

Component Reuse Phenomenon. Merullo et al. [27] have identified shared circuits for the IOI task and the Colored Objects task. We also observe this phenomenon in the factual recall task. As depicted in Table 4, we can observe that for related relations such as "city_in_country", "name_birth_place", and "country_language", their circuits include both L21H12, which stores and maps country-related information. Additionally, we found that some *relation heads* are activated by different relations. For instance, in our experiments, the head 'L7H14' appears in the circuits of both "official_language" and "official_currency". We speculate that these reused heads, rather than task-specific heads, can be considered topic heads, as proposed by [71, 60]. We believe that further investigation into this distinction is warranted in future research.

Table 4: Special component behaviour in circuits as task-specific head. Find more results in Appendix

Model	Type	Fact	Critical Component in Circuit
GPT2-Medium	Linguistic	Antonym	L17H2, L18H1, L13H12, L13H8
	Factual	city country	L21H12, L16H2
	Commonsense	work location	L19H15, L14H4, L13H3
	Bias	name country	L16H6, L21H12
GPT2-Large	Linguistic	Antonym	L25H5, L24H16, L19H13, L18H8
	Factual	company hq	L30H6, L25H13
	Commonsense	work location	L18H13, L28H18, L30H5
	Bias	name country	L21H19, L29H2
TinyLLAMA	Linguistic	Verb past tense	L17H0, MLP20
	Factual	Landmark country	L15H11, L17H19, MLP18
	Commonsense	Fruit Inside Color	L18H25, MLP18
	Bias	name country	L15H11, MLP17

B.3 A Case on TinyLLAMA

In this part, we demonstrate one circuit we found in the TinyLAMA in Figure 8. Actually, in TinyLLAMA, the attention heads bearing specific behaviors in the later layers is usually less than

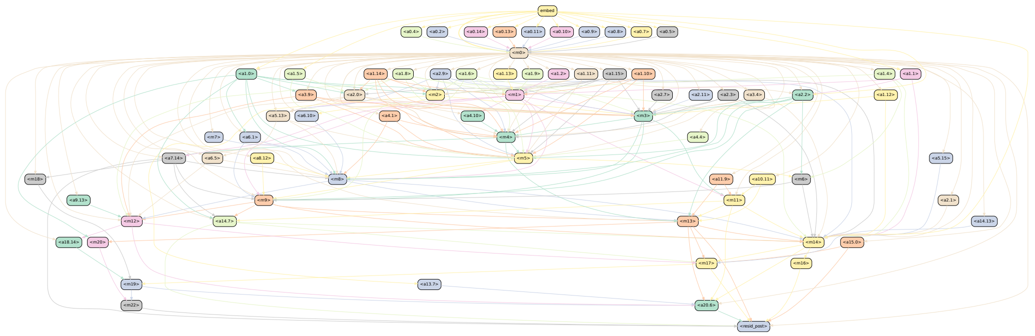


Figure 9: The knowledge circuit from the *"The official language of France is French"* in GPT2-Medium.

GPT2. We can also view some mover heads and relation heads in the circuit that would generate the target token as the output, such as L15H3 and L17H19.

C More Circuit Utilization Analysis

While previous work on knowledge storage suggests that knowledge may be localized in specific areas of the model, it is essential to ascertain whether the model actively employs this knowledge when encountering related contexts or if it relies on shortcuts.

C.1 Multi-hop Factual Knowledge Editing

We consider scenarios where the model makes a correct prediction for all the multi-hop questions and single-hop questions. We found a circuit reuse phenomenon in the one-hop and multi-hop knowledge circuits. Here, we first compute the proportion of the nodes N_{single} in the single-hop circuit C_{single} that appears in $N_{multiple}$ the set of nodes in the multi-hop circuit $C_{multiple}$.

$$Hit_{node} = \frac{|N_{multiple}| \cap |N_{single}|}{|N_{single}|} \tag{4}$$

Actually, there are two ways for the model to conduct multi-hop reasoning. As shown in the figure, the model can also answer the question by combining the two hop relations together (in the given case, combine "hometown" and "language" as "mother tongue") If the model is capable of combining two-hop relations in a more integrated or semantic way, such as inferring that the "mother tongue" is the language of one's hometown, this suggests a more complex reasoning process that goes beyond the simple overlap of nodes and edges. To capture this kind of reasoning, we assess the model's ability to integrate information from different hops. $R_{integrated}$ as the set of integrated relations (new paths created by combining information from different hops) We observe an overlap of these circuit nodes, indicating that the language model utilizes a large portion of the nodes in the original single hop's circuit, especially the mover head. From the overlap analysis, it seems the model utilizes single-hop information to conduct reasoning. We discover an intriguing phenomenon in GPT-2 and TinyLLAMA: the models can correctly answer first-hop knowledge and perform multi-hop reasoning based on it. However, interestingly, when we delete the first-hop knowledge while retaining only the second-hop relation and the first-hop subject, the models can still correctly answer the multi-hop question. Figure 10 illustrates a specific case in our findings. It further enhances our previous findings that the model actually conducts the factual recall with the relation head and the gathered information about the subject. **Moreover, we found this phenomenon is hugely alleviated by the aligned model, which requires further investigation in the future.**

Table 5: Hit for different hop

	First-hop	Second-hop	Integrate
node	83.33	70.27	71.42
edge	63.20	45.27	49.42

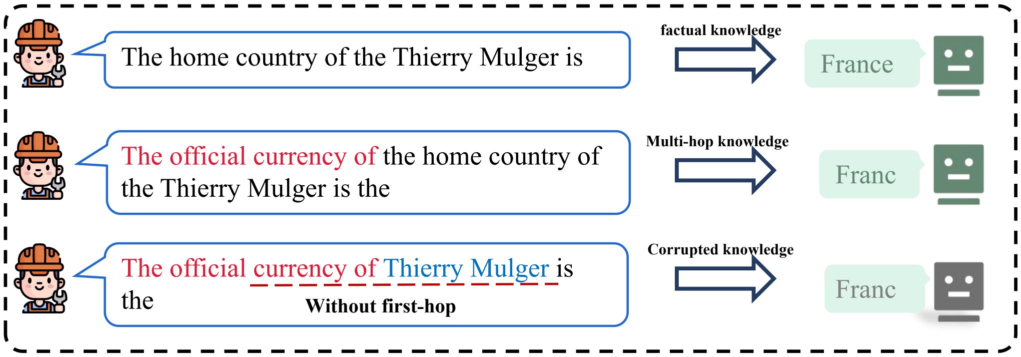

The home country of the Thierry Mulger is
factual knowledge
France
The official currency of the home country of the Thierry Mulger is the
Multi-hop knowledge
Franc
The official currency of Thierry Mulger is the
Without first-hop
Corrupted knowledge
Franc

Figure 10: a specific case in Multi-hop reasoning. When we removed the context of the first hop question, we found the model also directly gave us the answer. The phenomenon appears in both GPT2 and TinyLLAMA.

C.2 Hallucination

The results are presented in Figure 5. We observe an interesting phenomenon: the correct answer and the wrong answer are both accumulated at the previous layer, but at some specific layers, the wrong answer is selected as the answer to extract. We hypothesize that there may be a **circuit competition** here proposed by [72] and we detect the behavior of the specific component between them.

C.3 Reverse Relation

Reverse Curse [73] is an important issue in language models when models successfully give us the correct answer o for (s, r), but with a reverse relation $\hat{r}$ and o, the models fail to give us the correct subject s. In this part, we endeavor to investigate how the language model manages the reverse relation when they successfully store the knowledge. We first sample facts where the language model successfully predicts the given knowledge and the reversed fact. Then, we compute the overlap between these two circuits, $\mathcal{C}_d$ and $\mathcal{C}_r$ under node levels based on equation 4. We select the *"superhero_person"* relation to see the difference between these two circuits and test the node overlap of the two circuits in the model. We notice that the overlap between the two circuits is about 70%, indicating the language model may store the related information in the same place. We also found the activated mover heads for the two relationships to be identical.

D Edit Experiments

D.1 Method Implementation

ROME As proposed by Meng et al. [18], ROME views knowledge editing as a minimal optimization problem. ROME regards the MLP module as a simple key-value store. Specifically, the key represents a subject and the value encapsulates knowledge about that subject, the MLP can reestablish the association by retrieving the corresponding value for the key. To add a new key-value pair, ROME applies a rank-one modification to the MLP's weights, effectively "writing in" the new information directly. This method enables more direct and precise modification of the model's knowledge. The ROME method for model editing was conducted based on the EasyEdit[74] framework, utilizing the default parameters provided by EasyEdit. The experiments are performed on an A800 80G GPU, with approximately 8GB of memory consumption.

FT-M For Fine-Tuning (FT-M), we follow Zhang et al. [24]. It trains the MLP layer using the cross-entropy loss on the target answer while masking the original text. This approach aligns more closely with the traditional fine-tuning object. The FT-M method is conducted using the EasyEdit[4] [74] framework, with the default parameters provided by EasyEdit. The experiments are also performed on an A800 80G GPU, with a memory consumption of approximately 10GB.

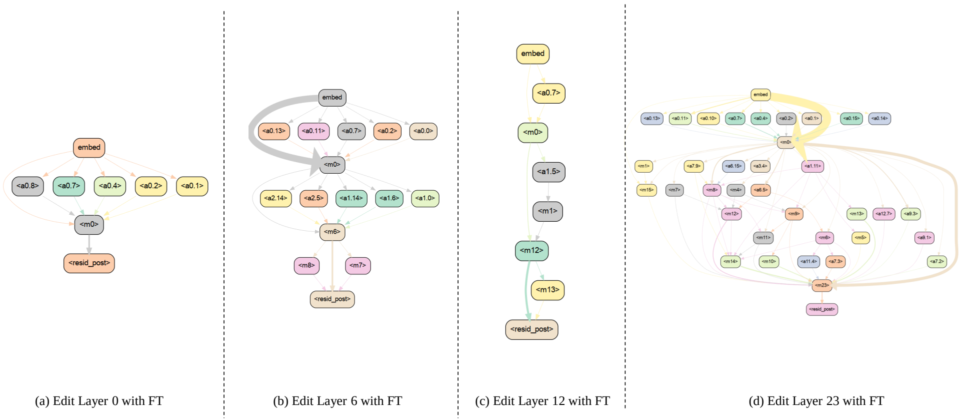
[4] https://github.com/zjunlp/EasyEdit

Table 6: Number of examples per relation and the count of accurate predictions by different LMs. This table is borrowed from Hernandez et al. [42] and here we sampled with different ways.

Category	Relation	#	# Correct in Hit@10		
			GPT2-Medium	GPT2-large	TinyLLaMA
factual	person mother	994	83	144	361
	person father	991	359	385	474
	person sport position	952	400	489	596
	landmark on continent	947	835	543	694
	person native language	919	310	220	260
	landmark in country	836	600	489	709
	person occupation	821	57	76	149
	company hq	674	308	312	470
	product by company	522	422	432	460
	person plays instrument	513	510	505	498
	star constellation name	362	266	148	297
	plays pro sport	318	317	316	315
	company CEO	298	20	52	125
	superhero person	100	28	35	50
	superhero archnemesis	96	6	6	23
	person university	91	14	37	35
	pokemon evolution	44	11	13	16
	country currency	30	25	25	30
	food from country	30	23	25	29
	city in country	27	20	23	27
	country capital city	24	24	24	24
	country language	24	24	24	24
	country largest city	24	24	24	24
	person lead singer of band	21	7	16	21
	president birth year	19	11	12	-
	president election year	19	17	18	-
commonsense	object superclass	76	62	64	72
	word sentiment	60	14	9	34
	task done by tool	52	44	45	45
	substance phase of matter	50	12	16	48
	work location	38	28	24	27
	fruit inside color	36	36	35	36
	task person type	32	28	27	26
	fruit outside color	30	16	20	21
linguistic	word first letter	241	236	235	241
	word last letter	241	135	73	114
	adjective antonym	100	80	81	84
	adjective superlative	80	24	19	63
	verb past tense	76	1	15	76
	adjective comparative	68	7	15	63
bias	occupation age	45	18	20	18
	univ degree gender	38	14	35	38
	name birthplace	31	29	30	31
	name religion	31	24	31	31
	characteristic gender	30	26	30	30
	name gender	19	19	19	19
	occupation gender	19	19	19	19

D.2 Edit Cases on FT-M and ROME

When a circuit is established for a particular piece of knowledge, we can manipulate the model's computation by targeting critical points within the circuit. Li et al. [75] ablates a small number of important causal pathways by masking the edges in the circuit and making the model less toxic and safer, which proves the effectiveness of the circuit. As illustrated in figure 12 and figure 13, we present the changes in the ranking of the predicted probabilities for the target new token when editing layer 6, 12, and 18 of the GPT-2 medium model using FT-M and ROME methods. When applying FT-M for model editing, it is evident that the rank of the target new token's probability sharply



(a) Edit Layer 0 with FT

(b) Edit Layer 6 with FT

(c) Edit Layer 12 with FT

(d) Edit Layer 23 with FT

Figure 11: The knowledge circuit obtained from the edited model for the case *"Platform Controller Hub was created by"* with the target entity "Intel" shows that when editing the model using different layers, the fine-tuned settings allow the edited MLP to directly provide the edited information.

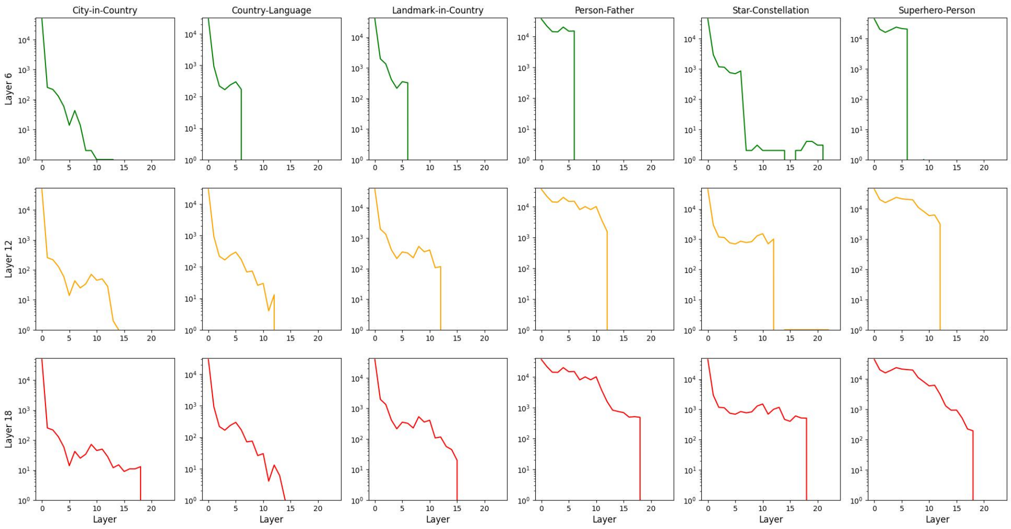


Figure 12: FT-M Rank Change Across Different Layers

declines at the corresponding edited layer, resulting in a vertical line in the figure. This indicates that FT-M directly embeds the editing information into the model's information flow. Conversely, when using the ROME method for editing, this effect is mitigated. The predicted probability of the target new token reaches its peak only a few layers after the edited layer. This observation is consistent with our previous analysis in Section 5.

E More related Work and Discussion

E.1 More Related Work

Knowledge in MLP Geva et al. [76] claim that MLP serves as key-value memories for knowledge in LMs. Geva et al. [77] further propose the knowledge neuron theory, suggesting that the key and

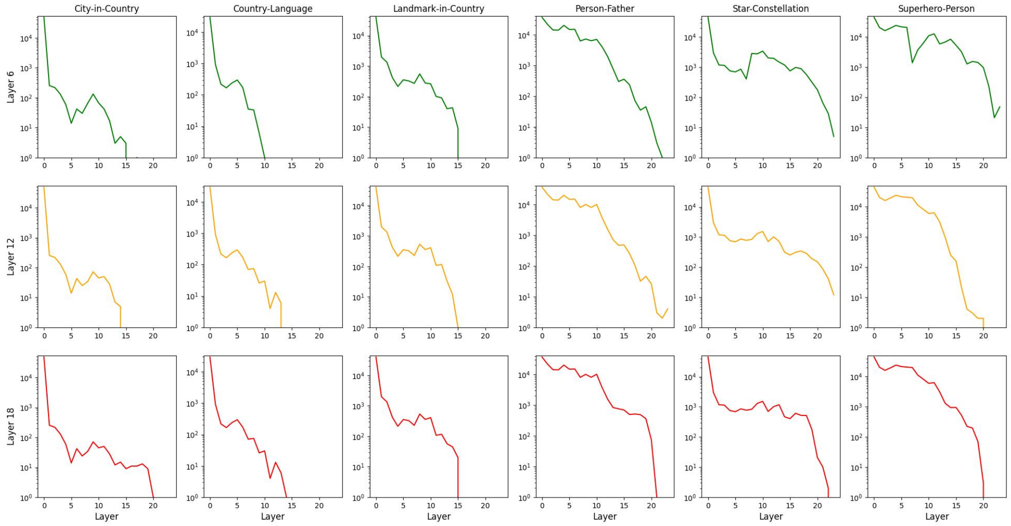


Figure 13: ROME Rank Change Across Different Layers

value vectors in MLPs encode factual knowledge. Based on the above findings, Chen et al. [78] observe that multiple distinct sets of KNs can store identical facts. Chen et al. [79] explore the structural and functional among neurons by neurological topology clustering method. Meng et al. [18] and Meng et al. [80] confirm that MLP modules do store factual knowledge and pioneering use of knowledge editing methods to modify outdated knowledge stored in language models. Anthropic recently introduces scaling monosemanticity[5], which extracts highly abstract features that respond to and behaviorally cause abstract behaviors.

Knowledge in Attention Heads Li et al. [58] reveal that some attention heads are capable of truthful answers. Wu et al. [81] investigate 4 model families, 6 model scales, and 3 types of finetuning, and find retrieval heads, which are responsible for retrieving relevant information from long context. Jin et al. [82] suggest that memory heads can retrieve knowledge from internal memory, while context heads can recall knowledge from external context. Todd et al. [83] use causal mediation analysis on a diverse range of in-context-learning and find some attention heads, dubbed function vectors, which trigger the ability of in-context-learning.

Knowledge in Hybrid Components Recent works emphasize the importance of connections of components among language models for knowledge representation and utilization. Geva et al. [13] describe factual recall by the following three steps: (1) subject enrichment in MLP sublayers, akin to ROME [18], (2) propagation of relations to the END token, and (3) selective extraction of attributes by attention heads in later layers. Lv et al. [31] apply projection and intervention to explore mechanisms in factual recalls tasks and conclude that task-specific attention head may move the topic entity to the final position of the residual stream, while MLP conducts relation function.

Circuit Circuit discovery also plays a significant role in analyzing the internal mechanisms of the entire model [25]. Specifically, a circuit, comprising components such as MLP and attention, is a subgraph of the computation graph. Conmy et al. [32] design an automated circuit discovery approach that implements the specified behavior. Wang et al. [26] explain the circuits for the Indirect Object Identification (IOI) task. They use causal interventions to discover circuits responsible for the flow of information. Instead the above task-specific circuit, Merullo et al. [27] presents evidence a circuit is shared by similar tasks in IOI and Color Object (CO) tasks. Dutta et al. [84] construct circuits using attention heads, and further observe that attention heads are pivotal to chain-of-thought reasoning, i.e., attention heads that move information along ontological relations exclusively appear in the initial half of the layers, while the tokens responsible for writing the answer predominantly appear in the later half of the layers. However, the above-mentioned circuit studies either focus solely

[5] `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`

on a single component (MLP or attention) or only explore IOI and CO tasks. IOI and CO tasks necessitate the model to search the preceding context for a matching token and then copy it into the next token prediction. Also, despite the success of previous circuit discovery, we can hardly make it into real usage. Hence, in this work, we attempt to analyze a knowledge circuit consisting of both MLP and attention components and investigate the effect of current editing methods on the circuit to shed light on the future.

Tools There are also many tools that are designed to analyze the LM's behavior, such as Logit Lens [85], Attention len [86], Attribution lens [42] and transformer-lens [41]. NeuroX [87] implements various interpretation methods under a unified API and provides insights into how knowledge is structured in representations and discovers the role of neurons in LM. Transformer Debugger [88] is an interpretability tool provided by OpenAI, which deploys the GPT-4 and sparse auto-encoder to explain the language neurons and attention head. PatchScope [89] is a tool provided by Google that uses a new model to explain the hidden states in the original model.

E.2 Limitation and Future Discussion

Despite of the attempt to combine the attention head and MLP to view the knowledge storage as a whole, this work operates with a relatively coarse granularity of circuits. For instance, the neurons within an MLP may necessitate a finer level of granularity to fully capture their behavior and contributions. Even though we now know these components work together to express the knowledge, why they are activated is still opaque. Our methodology employs the logit lens as a means to detect and analyze component information. However, this approach may encounter discrepancies between the middle layers and the output unembedding matrix. Such discrepancies can hinder a comprehensive and concrete analysis of the circuit components' behavior in the early layers. This limitation suggests the need for more robust techniques to bridge the gap between intermediate representations and final outputs. Recently, the Attention Lens method [86] has been proposed, which involves training a specific unembedding matrix to map each attention head into the vocabulary space. While this method is promising, it is also resource-intensive. Nevertheless, it represents a potential starting point for a deeper understanding of the knowledge circuits within neural models. Moreover, our research indicates that several mover heads are reused across different types of knowledge or relational contexts. The mechanisms by which these heads are activated and the conditions under which they operate require further exploration and may shed light on why neurons are sometimes "monosemantic" responding to a single feature, and sometimes "polysemantic" [90] responding to many unrelated features.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and Section 1 Introduction

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:Section 8 Limitations and Appendix E.2.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We report the setup throughout the paper as well as in the Subsection 3.3 and Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publicly available datasets (Appendix A), Code and Data are also provided in supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We report the setup throughout the paper as well as in the Subsection 3.3 and Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: [No]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer:[Yes]

   Justification: We use publicly standard datasets that do not contain information about individual people or offensive context to our knowledge. Ethical considerations are discussed in Section 8.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Section 8

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available artifacts and show them in Section 3.3.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.