Memory Injections: Correcting Multi-Hop Reasoning Failures during Inference in Transformer-Based Language Models

Mansi Sakarvadia[1,*], Aswathy Ajith[1], Arham Khan[1], Daniel Grzenda[1],
Nathaniel Hudson[1,2], André Bauer[1,2], Kyle Chard[1,2], Ian Foster[1,2]

[1]University of Chicago, [2]Argonne National Laboratory

Abstract

Answering multi-hop reasoning questions requires retrieving and synthesizing information from diverse sources. Large Language Models (LLMs) struggle to perform such reasoning consistently. Here we propose an approach to pinpoint and rectify multi-hop reasoning failures through targeted *memory injections* on LLM attention heads. First, we analyze the per-layer activations of GPT-2 models in response to single and multi-hop prompts. We then propose a mechanism that allows users to inject pertinent prompt-specific information, which we refer to as "memories," at critical LLM locations during inference. By thus enabling the LLM to incorporate additional relevant information during inference, we enhance the quality of multi-hop prompt completions. We show empirically that a simple, efficient, and targeted memory injection into a key attention layer can often increase the probability of the desired next token in multi-hop tasks, by up to 424%.

(a) Multi-hop prompt.

(b) Multi-hop prompt broken into 2 single-hop prompts.

Figure 1: A multi-hop prompt vs. two analogous single-hop prompts. The outputs are from GPT2-Small.

1 Introduction

Transformer-based *Large Language Models* (LLMs) (Vaswani et al., 2017; Brown et al., 2020) have shown exceptional promise for basic knowledge retrieval and language generation; however, they often lack the ability to perform basic reasoning tasks (Arkoudas, 2023; Guo et al., 2023; Blair-Stanek et al., 2023). In this work, we focus on the simple task of answering multi-hop prompts (i.e., prompts in which the subject is not stated explicitly), which humans handle easily but with which LLMs often struggle (see Fig. 1).

Researchers have attempted to rectify multi-hop reasoning failures by using various prompting methods such as *Chain-of-Thought* (CoT), *Tree-of-Thought* (ToT), and *Graph-of-Thought* (GoT) reasoning (Wei et al., 2022; Wang et al., 2023; Long, 2023; Xie et al., 2023; Yao et al., 2023; Besta
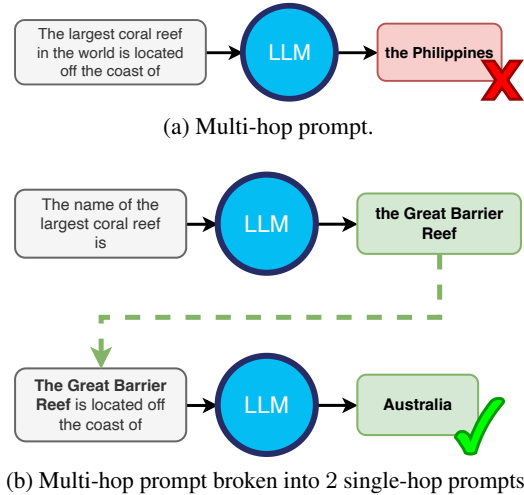
et al., 2023). However, these approaches often put the burden on users to know how to elicit desired responses—and, in the hands of non-expert users, can lead to unreliable prompt completions. Researchers have also proposed model editing (Meng et al., 2022a,b; Zhong et al., 2023; Li et al., 2023) approaches that may hard-code distant relationships directly into model weights, rather than enhancing the model's abilities to recall and then link simpler relationships. These approaches can be computationally expensive and have unintended effects on other knowledge originally embedded in the model's weights (Cohen et al., 2023).

Our approach to this problem is based on the hypothesis that LLMs often fail to recall relevant *memories* when attempting to answer a prompt that requires multiple "hops" of reasoning, rather than lacking knowledge of the *memories* altogether. For example, when attempting to complete the multi-hop prompt, "The largest coral reef system in the world is located off the coast of...," we hypothesize that the model does not correctly recall that "the largest coral reef system in the world"

---

is "the Great Barrier Reef" before predicting the next token in the sequence. Yet the model can accurately complete both the corresponding single-hop prompt "The Great Barrier Reef is located of the coast of...," and, when prompted, "the largest coral reef" as "the Great Barrier Reef." Clearly, this information was encoded in the model during training but is not incorporated when answering questions that reference the prompt's subject indirectly. In this case, therefore, we define the missing *memory* to be "the Great Barrier Reef."

To study our hypothesis, we first attempt to reverse engineer a key mechanism by which transformer-based LLMs conduct reasoning. Specifically, we find that in transformer-based models it is attention heads, rather than multi-layer perceptrons, that are responsible for retrieving *memories* critical to successful model predictions; our finding is further substantiated by similar findings by Li et al. (2023); Geva et al. (2023); Dar et al. (2022). We then study instances in which this mechanism fails in multi-hop reasoning tasks and find that this mechanism is likely the source of incorrect, insufficient, or irrelevant *memory* retrievals **(Contribution 1)**—for an example, see Fig. 2.

We then propose a lightweight *memory injection* method that can be employed to correct a multi-hop reasoning failure during inference **(Contribution 2)**. As an example: by employing our method to inject the *memory* of "The Great Barrier Reef" into the multi-hop prompt "The largest coral reef system in the world is located off the coast of..." during inference, we increase the probability of the next token "Australia" by 189%; refer to Fig. 3 for details.

For our analyses, we hand-crafted a dataset for interpretabilty purposes **(Contribution 3)** and make use of a larger programmatically-generated dataset—refer Table 1 for more information.

Finally we conduct additional experiments **(Contribution 4)** to:

1. Identify the ideal layer and magnitude for the *memory injection*.

2. Demonstrate the significance of curating prompt-specific *memories* for injection.

3. Analyze if *memories* drawn from different parts of speech—namely, nouns, adjectives, adverbs, conjunctions, verbs—behave differently during *memory injection*.

2 Background & Notation

We define single- vs. multi-hop prompts and provide a formal definition of the transformer model.

2.1 Multi-hop vs. single-hop prompts

We refer to a prompt as *single-hop* if the subject of the relation is stated explicitly in the prompt, and *multi-hop* otherwise. Multi-hop prompts refer to their subject in a way that requires an additional "hop" or inference step. For example, consider the single-hop prompt, "*George Washington* fought in the..." with a correct answer being "Revolutionary War." In the analogous multi-hop prompt, "*The first president of the United States* fought in the...," a preliminary inference step is needed to identity of the first US president before predicting the next token. For additional examples of single- and mutli-hop prompts, see Table 3 in the appendix.

2.2 Transformer Architecture

We introduce a common notation for the components of the transformer-based language model architectures that are the focus of our analyses. Specifically, we focus on auto-regressive, decoder-only models. We adopt much of our notation from Elhage et al. (2021) and Geva et al. (2023).

2.2.1 Embedding Inputs

An input text is parsed into N distinct tokens $t_0, \cdots, t_N$. Each token t_i is then embedded as $x_i^0 \in \mathbb{R}^d$ via an embedding matrix $W_E \in \mathbb{R}^{|V| \times d}$, where V is the vocabulary and d is the hidden dimension.

2.2.2 Residual Stream

Following the embedding layer, all tokenized embeddings x_i^0 are passed through a series of *residual blocks*. The outputs of each *residual block* are added back into the model's *residual stream* denoted by R^ℓ ($\forall \ell \in \{1, \cdots, L\}$) where L is the number of layers in the LLM.

We define the *residual stream* at layer ℓ as:

$$R^\ell = [x_0^\ell, \cdots, x_N^\ell], \qquad (1)$$

where x_i^ℓ is the representation of token i at layer ℓ. The *residual stream* is updated by its respective *residual block* r^ℓ:

$$R^{\ell+1} = R^\ell + r^{\ell+1}, \qquad (2)$$

and the output of a *residual block* r^ℓ is:

$$r^\ell = a^\ell + m^\ell, \qquad (3)$$

where a^ℓ is the output of the *Multi-Headed Self Attention* (MHSA) layer and m^ℓ is the output of the *Multi-Layer Perceptron* (MLP). We define MHSA and MLP in the following sections.

2.2.3 Multi-Headed Self Attention (MHSA)

Each MHSA layer ℓ is defined via four parameter matrices $W_Q^\ell, W_K^\ell, W_V^\ell, W_O^\ell \in \mathbb{R}^{d \times d}$ ($\forall \ell \in \{1, \cdots, L\}$) and the hyperparameter H denotes the number of attention heads. Following Elhage et al. (2021) and Geva et al. (2023), we can further dissect our parameter matrices to better observe the relationship between unique sets of parameters and individual attention heads: $W_Q^{l,j}, W_K^{\ell,j}, W_V^{\ell,j} \in \mathbb{R}^{d \times \frac{d}{H}}$ and $W_O^{\ell,j} \in \mathbb{R}^{\frac{d}{H} \times d}$ for $j \in [1, H]$. Now, we can define the output of each MHSA a^ℓ as the sum of all attention head outputs,

$$a^\ell = \sum_{j=1}^{H} h^{\ell,j}, \qquad (4)$$

where $h^{\ell,j}$ is the output of the jth head in layer ℓ:

$$h^{\ell,j} = A^{\ell,j}\big(R^{\ell-1}W_V^{\ell,j}\big)W_O^{\ell,j}. \qquad (5)$$

$$A^{\ell,j} = \text{softmax}\left(\frac{\big(R^{\ell-1}W_Q^{\ell,j}\big)\big(R^{\ell-1}W_K^{\ell,j}\big)^T}{\sqrt{d/H}} \odot M\right) \quad (6)$$

where the $\text{softmax}(\cdot)$ is performed as a row-wise operation, $\odot$ is the Hadamard product, and $M \in \{0,1\}^{N \times N}$ is an auto-regressive attention mask where masked token positions are set to 0.

2.2.4 Multi-Layer Perceptron (MLP)

Each MLP is defined via two parameter matrices $W_F^\ell, W_I^\ell \in \mathbb{R}^{d \times d_p}$ with inner-dimension d_p and a nonlinear activation function, σ.

$$m^\ell = W_F^\ell \, \sigma\Big(W_I^\ell\big(a^\ell + R^{\ell-1}\big)\Big) \qquad (7)$$

2.2.5 Unembedding Predictions into Logits

After the final *residual block*, all token positions x_i^{-1} will be projected back into the vocabulary domain via the *unembedding matrix* $W_U \in \mathbb{R}^{d \times |V|}$. The output of the last token position is the next token prediction of the model.

3 Experimental Overview

Our central aim is to better understand how the outputs of the attention heads affect model performance with respect to predicting the correct next token in prompts requiring single-hop reasoning versus in prompts requiring multi-hop reasoning.

3.1 Dataset Descriptions

We employ three datasets in this work. Two, used to assess model prompt completion accuracy, are our own high-quality manually curated dataset of single and multi-hop pairs and a programmatically generated dataset of prompt pairs. The third comprises lists of words from common parts of speech, which we use to study how the effectiveness of our intervention varies with the part of speech of injected tokens.

3.1.1 Programmatically Generated Dataset

The 2WikiMultiHop dataset (Ho et al., 2020) contains pairs of knowledge triples $\{(s_1, r_1, s_2)_1, (s_2, r_2, s_3)_2\}$, each with two subjects s and a relationship r. We used these knowledge triples, plus a set of predefined templates, to generate a set of pairs of single- and multiple-hop questions, *2WMH*: see Tables 1 and 3.

For example, let s_1 = "Lilli's Marriage," r_1 ="director," s_2 = "Jaap Speyer," r_2 = "country of citizenship," s_3 = "Dutch." Then for **single-hop**, the template: "The r_2 of s_2 is ... s_3", the prompt yields the prompt "The country of citizenship of Jaap Speyer is ... [Dutch]"; for **multi-hop**, the template "The r_2 of the r_1 of s_1 is ... s_3" yields then the prompt: "The country of citizenship of the director of Lilli's Marriage is ... [Dutch]."

3.1.2 Human-Generated Dataset

As evidenced by the example presented above, the *2WMH* dataset, while scalable, contains many grammatical flaws. Therefore, we construct an additional dataset for multi-hop reasoning with a focus on grammatical and factual correctness presented below. We hand-crafted 106 (single-hop, multiple-hop) prompt pairs, each in the same form as those in *2WMH*: e.g., **single-hop**: "St. Peter's Basilica is in the city of... [Rome]" and **multi-hop:** "The biggest church in the world is in the city of... [Rome]". Each prompt pair was also evaluated by two external reviewers for factual and grammatical accuracy. We hereafter refer to this dataset as *Hand*; see Tables 1 and 3.

3.1.3 Part of Speech Dataset

We used a subset of the Corpus of Contemporary American English (Davies, 2011) which compiles

344

Data	Size	Model	Single-hop			Multi-hop		
			Answer prob.	Surprisal	Prompt len.	Answer prob.	Surprisal	Prompt len.
Hand	106	GPT2-Small	0.157	4.21	9.66	0.087	4.91	12.99
Hand	106	GPT2-Large	0.28	2.90	9.66	0.157	3.97	12.99
2WMH	1000	GPT2-Small	0.0007	9.80	10.44	0.00086	9.64	14.00
2WMH	1000	GPT2-Large	0.0023	8.71	10.44	0.002	8.57	14.00

Table 1: Properties of the datasets used in our work. *Size*: Number of prompts. *Answer prob.*: Average model probability model for expected next token. *Surprisal*: Average model surprisal value for expected next token ($surprisal \triangleq -\log(p)$ where p is a probability). *Prompt len.*: Average tokenized length of prompt.

word frequencies (Davies, 2010) to generate lists of (i) the most common words from various parts of speech: 824 adjectives, 331 adverbs, 40 conjunctions, 2635 nouns, 969 verbs, and (ii) the 5050 most common words overall ("top 5050").

3.2 Model Description

We work with two pretrained GPT2 models (Radford et al., 2019). **GPT2-Small** has 12 layers, 12 attention heads per attention layer, and ∼160M parameters. **GPT2-Large** has 36 layers, 20 attention heads per attention layer, and ∼840M parameters. Both have a vocabulary of ∼50K tokens.

3.3 Tools & System Setup

We use the *Transformer Lens* Python package (Nanda and Bloom, 2022) to cache, inspect, and construct interventions on model inference passes. We ran experiments on a single A100 GPU with 40 GB RAM. Experimental code, dependency information, and datasets are available on GitHub.[1]

4 Proposed Methods

Recent work suggests that attention heads are knowledge retrievers during a model's inference pass (Geva et al., 2023; Li et al., 2023). Extending this result to multi-hop prompts, we hypothesize that attention layers play an important role in retrieving memories relevant to the "hop" in a given prompt. Therefore we define two algorithms below: one for analyzing attention head outputs in embedding space and the other for injecting a targeted memory into a model's hidden activations in order to correct faulty/incomplete reasoning.

4.1 Interpreting Attention Heads

We want to further understand the outputs of individual heads, and more specifically assess if any individual attention heads are exercised differently by single-hop vs. multi-hop prompts.

Inspired by Logit Lens (nostalgebraist, 2021), we leverage the model's unembedding matrix to study the internal mechanism of each attention head. For attention head j in layer ℓ, $h^{\ell,j}$, we apply the model's unembedding matrix W_U followed by a $\text{softmax}(\cdot)$ operation and interpret the last token position (out of N total tokens) as a set of probabilities over tokens in the vocabulary space:

$$vocab^{\ell,j} = \text{softmax}(h^{\ell,j}W_U)_{N-1} \qquad (8)$$

See in Fig. 2 an example of discrepancy in attention head behavior, when using Eq. (8), for analogous single vs. multi-hop prompts. See additional examples in Table 5.

A potential limitation of this approach is that it may portray attention head behavior inaccurately due to representational drift between model layers—and, like (nostalgebraist, 2021), may not generalize to other models. Nevertheless, we find it to be an effective preliminary tool for studying the function of attention heads in updating the output distribution. We leave the development of an interpretability tool that considers these drawbacks to future work.

4.2 Memory Injections to Correct Failures

Fig. 2 shows how Eq. (8) can reveal discrepancies between attention head behaviors for single- vs. multi-hop prompts. We hypothesize that such discrepancies arise because the model, when updating the output distribution in each layer, fails to incorporate information about the implicit entity in the multi-hop prompt. This seems reasonable, as to retrieve information about an implicit entity one likely must first relate that entity to some explicit subject and then retrieve relevant information (hence our notion that processing prompts with implicit subjects requires an extra hop compared to those with explicit subjects).
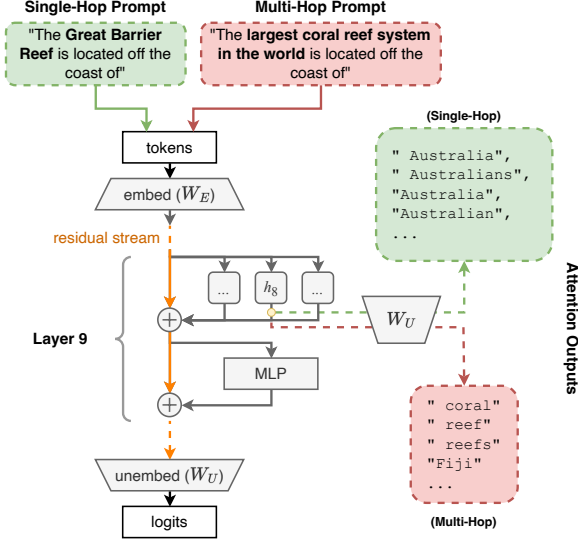
---

[1] https://github.com/msakarvadia/memory_injections

345

Figure 2: **Diagram of language model reasoning.** Highest ranked attention outputs of GPT2-Small at layer $\ell = 9$, head $h = 8$ when projected into vocabulary space (via the GPT2-Small embedding matrix) for a single-hop prompt (green) and its multi-hop counterpart (red).

Thus we design a method (see Fig. 3) for injecting a missing hop directly into the output hidden states of an attention head before those outputs are added back into the transformer's residual stream:

1. Let m be a memory (a phrase, for example: "The Great Barrier Reef") and let τ be the magnitude of the memory injection.

2. Tokenize the memory m into $t_0, \cdots, t_q$ where q is the number of tokens. We encode each token t_i into a one-hot vector $b_i \in \{0, 1\}^{|V|}$ and sum all resulting one-hot vectors b_i together into a binary vector $B \triangleq \sum_i b_i$.

3. Embed the binary vector, B, back into the model's latent space by applying the transpose of the unembedding matrix:

$$B^* = B\, W_U^T \qquad (9)$$

4. Then, to inject a memory at the attention layer of layer ℓ, add the embedded memory into the outputs of the attention heads during the inference pass:

$$a^\ell = \sum_{j=1}^{H} h^{\ell,j} + \tau B^* \qquad (10)$$

See additional examples of *memory injections* in Table 4.

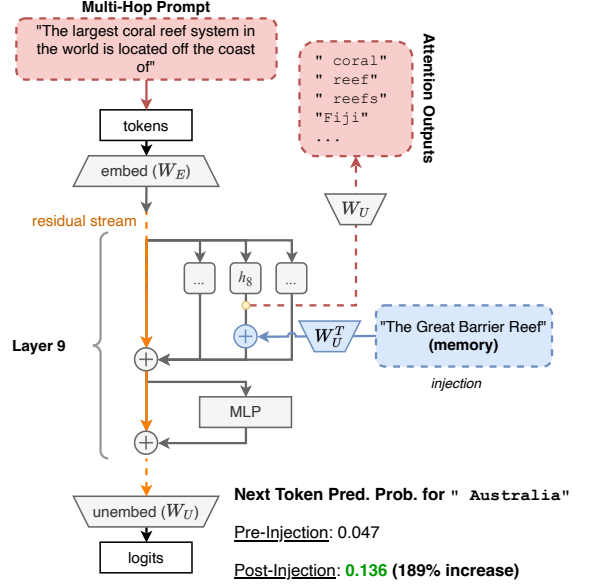Figure 3: **Memory injection.** Injecting memory "The Great Barrier Reef" into GPT2-Small hidden activations at layer $\ell = 9$, head 8, $\tau = 4$.

5 Results and Discussion

We report, in turn, on our curated memory, random memory, and part-of-speech injection experiments.

5.1 Curated Memory Injections

We hypothesize that a model's poor performance on multi-hop prompts is due to its inability to resolve the implicit subject (e.g., "The largest coral reef system in the world") to an explicit subject (e.g., "The Great Barrier Reef"). This failure limits the later layers' ability to retrieve relevant information about this subject before predicting the next token. Therefore, in this experiment, we curate sets of tokens to inject into our model's residual stream such that it can resolve the explicit subject more easily. We further study the effect that the injection magnitude τ has on its success.

Experimental design: For every multi-hop prompt in our datasets, we extract the explicitly stated subject from the corresponding single-hop prompt and inject those tokens as *memories* into each attention layer as described in Section 4.2. For example, given the **single-hop prompt** "*The Great Barrier Reef* is located off the coast of…" and the **multi-hop prompt** "*The largest coral reef system in the world* is located off the coast of…," the **memory** is "*The Great Barrier Reef.*"

We assess the effects of injection layer ℓ and magnitude $\tau \in [1, \cdots, 15]$ by enumerating the resulting change in accuracy for all combinations
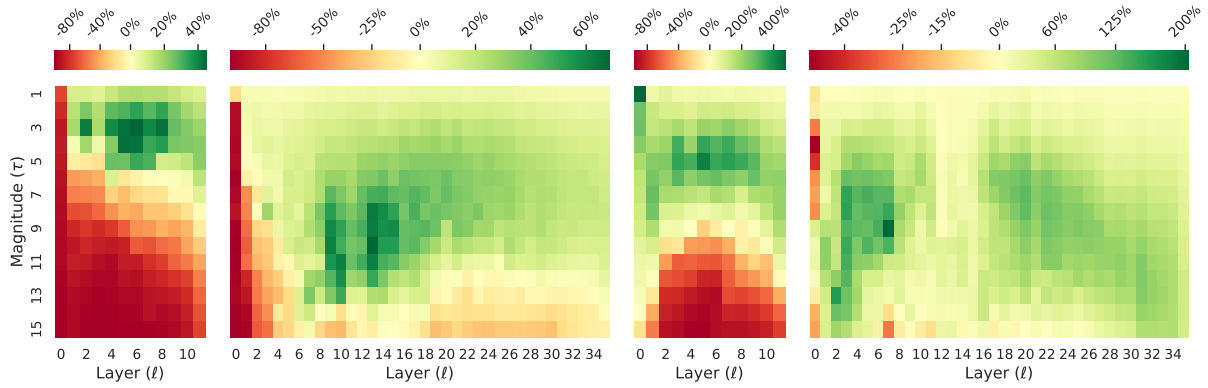
Figure 4: **Curated memory injections.** From left to right: GPT2-Small + *Hand*, GPT2-Large + *Hand*, GPT2-Small + *2WMH*, GPT2-Large + *2WMH*. Each cell in each heatmap is the average percent difference between the pre- and post-injection next token predictions for multi-hop prompts. Green cells denote a positive percent difference (i.e., correct prediction is more likely), while red cells denote a negative percent difference (i.e., correct prediction is less likely). When computing the averages for each (ℓ, τ) pair we exclude outliers not within ± 2 standard deviations from the mean.

of these two parameters for both GPT2-Small and GPT2-Large. We measure the success of a memory injection by calculating the percent increase between the model's predicted probability for the expected next token from the multi-hop prompt with and without the injection. A greater positive difference indicates a more successful injection.

Discussion: Results are in Fig. 4. We observe that each model/dataset combination has an optimal layer ℓ and magnitude τ for memory injections: the darkest green areas, which signify the highest average percent increase in probability of the expected next token for the respective dataset. The best (ℓ, τ) pair injection results are in Table 2. Additional examples of memory injections are in Table 4.

5.2 Random Memory Injections

In Section 5.1, we identify ideal (ℓ, τ) pairs for each model and dataset for a curated memory injection. We now demonstrate that the results we observe are not spurious: i.e., the information that we inject at each head should be related to the explicit subject. We demonstrate the need for our particular injection routine by assessing the effects on model accuracy of randomly injecting tokens from various parts of speech.

Experimental design: We conduct targeted injections for the high-scoring (ℓ, τ) pairs identified via the experiment in Section 5.1, Table 2. Instead of injecting curated subject tokens, we select as candidate injections the 40 most common words from each of the adjectives, adverbs, conjunctions, nouns, verbs, and top 5050 subsets of our *Part of*

Speech dataset. We then apply each word as an individual injection for every prompt in our multi-hop dataset at the ideal (ℓ, τ) pair. We term these injections "random," as they were not curated to be relevant to our prompts.

Discussion: The results are in the right half of Table 2. We observe that a random injection led, on average, to a degradation in predictive performance across most parts of speech considered, as indicated by a negative percent difference (decrease in correct answer probability) between the pre- and post-injection expected next token probabilities for multi-hop prompt completions. Additionally, no random injection result exceeded the performance of a curated injection. These findings suggest that the choice of injected tokens is critical for improving multi-hop prompt completion success.

5.3 Memory Injections for Parts of Speech

We have tested curated vs. random memory injections at ideal (ℓ, τ) pairs. Now we assess whether memory injections from specific parts of speech more broadly have positive impacts on prompt completions, not just at the ideal locations for curated memories, but also at other (ℓ, τ) pairs. Our hypothesis is that if a transformer-based LLM has learned a division of labor regarding which attention layers are responsible for retrieving specific concepts (e.g., parts of speech) then this experiment might highlight those learned roles.

Experimental design: This experiment is identical to that of Section 5.1, except that: (i) for each part of speech $pos \in$ [adjectives, adverbs, conjunc-

| Model | Data | ℓ | τ | Curated | Random | | | | | |
				Subject	Adj.	Adv.	Conj.	Noun	Verb	Top-5050
GPT2 Small	Hand	7	3	**45%**	-7.6%	-6.0%	-6.3%	-6.5%	-7.5%	-6.0%
GPT2 Small	2wmh	6	5	**424%**	-17.1%	-15.1%	-10.3%	-1.1%	-1.2%	1.6%
GPT2 Large	Hand	14	10	**68%**	-8.1%	-4.4%	-4.9%	-9.8%	-6.0%	-4.7%
GPT2 Large	2wmh	8	9	**204%**	13.0%	11.6%	3.5%	11.8%	4.3%	17.6%

Table 2: **Curated vs. random memory injections.** Table shows the (ℓ, τ) pairs for the best token injections, along with the *average percent difference* (excluding outliers >±2 standard deviations from the mean) between pre- and post-injection expected next token predictions for multi-hop prompts. Each random injection column indicates 40 random injections from [Adjectives, Adverbs, Conjunctions, Nouns, Verbs, Top 5050] at the ideal (ℓ, τ).
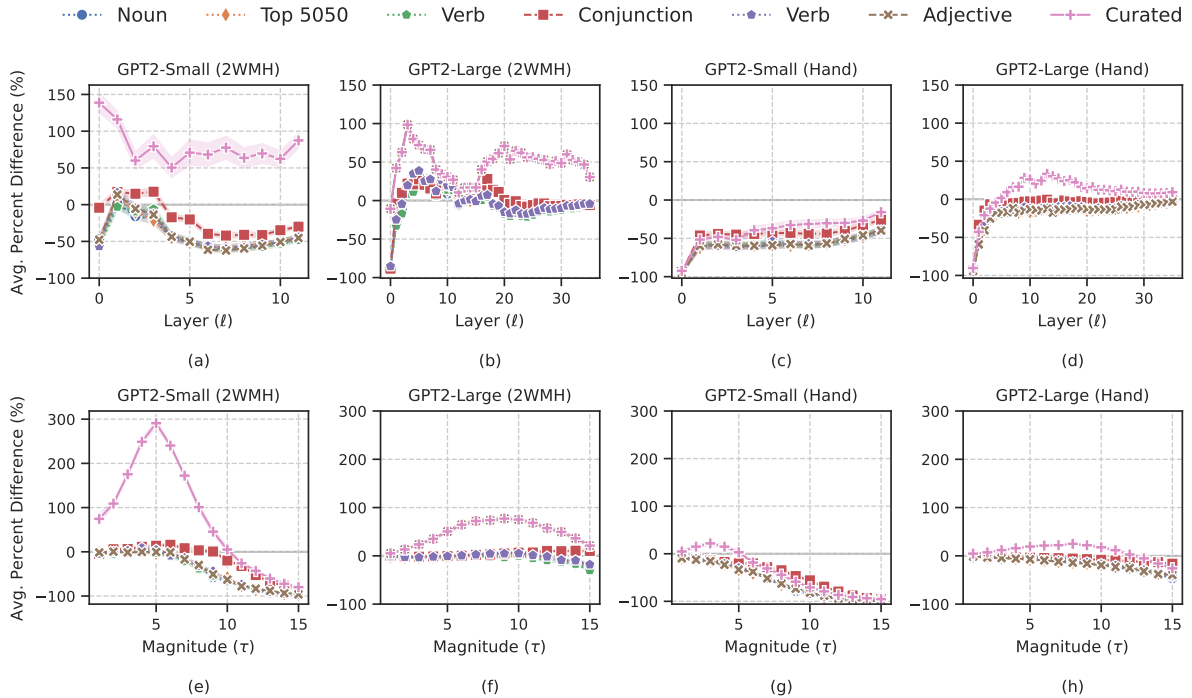

Figure 5: **Part of speech memory injections.** This figure shows the average effect of memory injections from various parts of speech as a function of layer ℓ (top row) and magnitude τ (bottom row). The standard deviation scaled by 10% is pictured across magnitudes (top row) and layers (bottom row).

tions, nouns, verbs, top 5050], we use a randomly selected word: e.g., "apple" from "nouns"; and (ii) when searching for the ideal (ℓ, τ) pair for a given part of speech and multi-hop prompt, we use a new random word for each injection.

Discussion: The results are in Fig. 5. We note that for no part of speech considered here does the average performance of the studied memory injections exceed that of the curated memory injections presented in Table 2. Additionally, memory injections from adjectives, adverbs, nouns, verbs, and top 5050 seemed to exhibit similar behavior. Memory injections from conjunctions, however, typically outperformed all other parts of speech. We hypothesize that this is because conjunctions often play a neutral role in prompt completions.

Thus, while a random noun (e.g., "apple") might distort prompt completion, a random conjunction (e.g., "and," "for") is less likely to do so.

We note also that for each part of speech, performance averaged over all injections for most (ℓ, τ) pairs was reduced (< 0) for *Hand* (refer Fig. 5: subplots c, d, g, h), but was sometimes improved (> 0) for *2WMH* (refer Fig. 5: subplots a, b, e, f). We attribute this result to the relative difficulties of the two datasets. *Hand* has, on average, lower surprisals than does *2WMH*, as seen in Table 1, suggesting that there is additional information that the model could use successfully for *2WMH*, but not for *Hand*.

These results (see also the Appendix; Figs 6–9) suggest that while curated memories are ideal for

correcting multi-hop reasoning failures, language models can also benefit from injections of different parts of speech. This result suggests that different parts of a language model (namely, early layers) serve specialized roles, with some dealing with processing related to specific parts of speech.

In future work we will curate relevant memories from various parts of speech for each prompt, to better understand the effects of curated memories.

6 Related Work

Much recent work has focused on the inner workings of Transformers (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Radford et al., 2019). Nanda et al. (2023) explore how the emergent properties of LLMs form during training. Recent interpretability research has focused on the mechanisms by which linear layers in LLMs retrieve information, characterizing them as key-value stores of information (Geva et al., 2021; Dai et al., 2022a,b) and showing that tokens can be characterized by their distribution in the output vocabulary (Geva et al., 2022).

Others have also examined the intermediate activations of LLMs in order to uncover underlying reasoning mechanisms. nostalgebraist (2021) applied GPT-2's unembedding matrix to intermediate layers to interpret how the model arrives at its final answer. Belrose et al. (2023) employed a learned transformation to mitigate the effect of any bias introduced by using the unembedding matrix.

There has been much recent interest in whether LLMs are reliable stores of information for attempting to both identify where knowledge exists and how to edit stored factual knowledge effectively (Mitchell et al., 2022a,b; Elazar et al., 2021; Hase et al., 2023). Recent approaches to knowledge editing make use of learned hyper-models to edit weights, additional trained parameters, or direct interventions on model weights (De Cao et al., 2021; Huang et al., 2023; Dhingra et al., 2022). However, these approaches raise another issue: dealing with knowledge retention and preventing catastrophic forgetting (Jang et al., 2022; Hase et al., 2021; Zhong et al., 2023). Additionally, it is not clear that the mechanisms by which model predictions are constructed is fully understood, limiting our ability to improve model performance (Turpin et al., 2023). Some approaches propose to use external knowledge stores such as knowledge graphs to augment the factual capabilities of LLMs (Jiang et al., 2023; Sun et al., 2018; Zhang et al., 2022).

7 Conclusions and Future Directions

We demonstrate that a key reason LLMs perform worse on multi-hop prompts is because they fail to recall intermediary information that is relevant to a hop. We find that attention heads play an important role in this factual recall process, and that in the case of multi-hop reasoning, certain attention layers fail to recall relevant information. To rectify this shortcoming, we establish an algorithm for injecting "memories" directly into the model's hidden activations during inference. Through experimentation, we find that injecting relevant memories into the hidden activations of the attention heads during inference is an efficient way to boost model performance on multi-hop prompts.

We anticipate that our memory injection scheme can extend a model's longevity by enabling less frequent retraining/fine-tuning. We also hope in future work to demonstrate the use of memory injections to correct stale or incorrect information, remove private or harmful information, and combat bias during LLM inference.

There is also a tremendous opportunity to scale online-memory injections to enhance the quality of thousands/millions of model inferences, if we can automate the process of memory selection via unsupervised algorithms, for instance by connecting LLMs with knowledge bases.

Limitations

Internal biases of the question writers as well as the rigid structure that had to be imposed on the prompt structure mean that our human-generated dataset is representative only of a small fraction of the many types of multi-hop questions. Furthermore, our hand-generated dataset is relatively small compared to our programmatically generated dataset. Additionally, our analyses were limited to GPT2-Small and GPT2-Large; further work is needed to determine whether, as we expect, other language models sharing a transformer-based architecture and a similar unsupervised causal language modeling training objective display similar behavior. Lastly, we rely on the model's unembedding matrix W_U to interpret model hidden states and embed *memories* for injection. While for our work, results indicate that this transformation was sufficient, we acknowledge that this unembedding matrix is not tuned to interpret intermediate layers;

we aim to address this shortcoming in future work by instead using layer-specific learned projections to transform between hidden states and vocabulary.

Ethics

Our attention head inspection mechanism uncovered several sources of bias (such as racism); refer Table 5 for examples. We expect a more detailed study of the attention heads of GPT2-Small and GPT2-Large, as well as other LLMs, to reveal additional undesirable behaviors. We aim in future work to use our inspection method to uncover (and hopefully address) these biases.

Acknowledgements

References

Konstantine Arkoudas. 2023. GPT-4 can't reason. *arXiv preprint arXiv:2308.03762*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can GPT-3 perform statutory reasoning? *arXiv preprint arXiv:2302.06100*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *arXiv preprint arXiv:2307.12976*.

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

Damai Dai, Wenbin Jiang, Qingxiu Dong, Yajuan Lyu, Qiaoqiao She, and Zhifang Sui. 2022b. Neural knowledge bank for pretrained transformers. *arXiv preprint arXiv:2208.00399*.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2022. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*.

Mark Davies. 2010. The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*, 25(4):447–464.

Mark Davies. 2011. Word frequency data from the Corpus of Contemporary American English (COCA).

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

N Elhage, N Nanda, C Olsson, T Henighan, N Joseph, B Mann, A Askell, Y Bai, A Chen, T Conerly, et al. 2021. A mathematical framework for transformer circuits.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. 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*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Taicheng Guo, Kehan Guo, Zhengwen Liang, Zhichun Guo, Nitesh V Chawla, Olaf Wiest, Xiangliang Zhang, et al. 2023. What indeed can GPT models do in chemistry? a comprehensive benchmark on eight tasks. *arXiv preprint arXiv:2305.18365*.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? Surprising differences in causality-based localization vs. knowledge editing in language models. *arXiv preprint arXiv:2301.04213*.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. Do language models have beliefs? Methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun KIM, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. PMET: Precise model editing in a transformer. *arXiv preprint arXiv:2308.08742*.

Jieyi Long. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Neel Nanda and Joseph Bloom. 2022. Transformer-Lens.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

nostalgebraist. 2021. Logit Lens on non-GPT2 models + extensions.

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

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Decomposition enhances reasoning via self-evaluation guided decoding. *arXiv preprint arXiv:2305.00633*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784, Dublin, Ireland. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

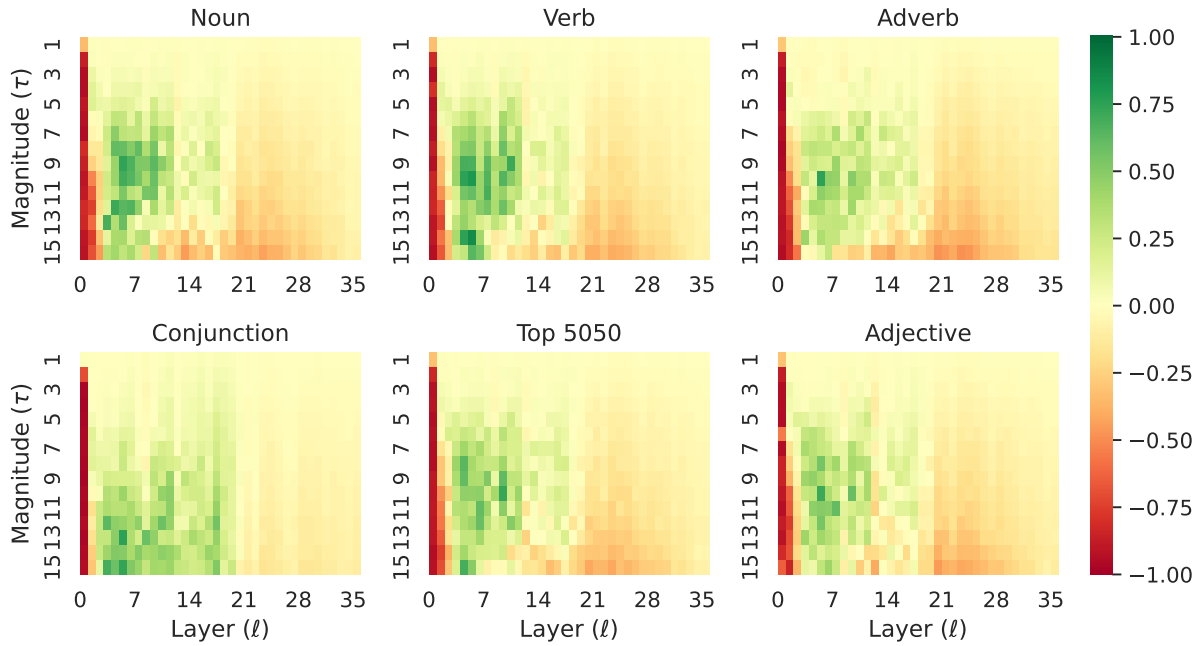# A   Part-of-Speech Memory Injection Appendix

Figure 6: **GPT2-Large, *2WMH* dataset.** Heatmap shows average percent difference between pre- and post-injection answer probabilities for multi-hop prompts excluding outliers not within ± 2 standard deviations from the mean across various parts of speech.

Figure 7: **GPT2-Large, *Hand* dataset.** Heatmap shows average percent difference between pre- and post-injection answer probabilities for multi-hop prompts excluding outliers not within ± 2 standard deviations from the mean across various parts of speech.
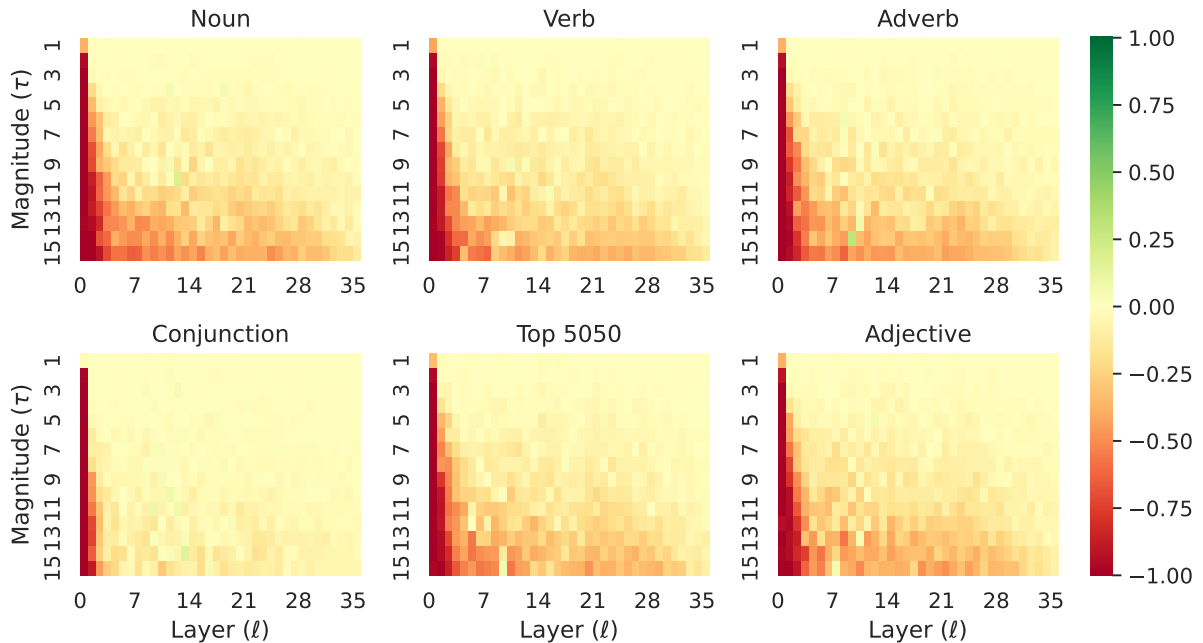
Figure 8: **GPT2-Small, *2WMH* dataset.** Heatmap shows average percent difference between pre- and post-injection answer probabilities for multi-hop prompts excluding outliers not within ± 2 standard deviations from the mean across various parts of speech.
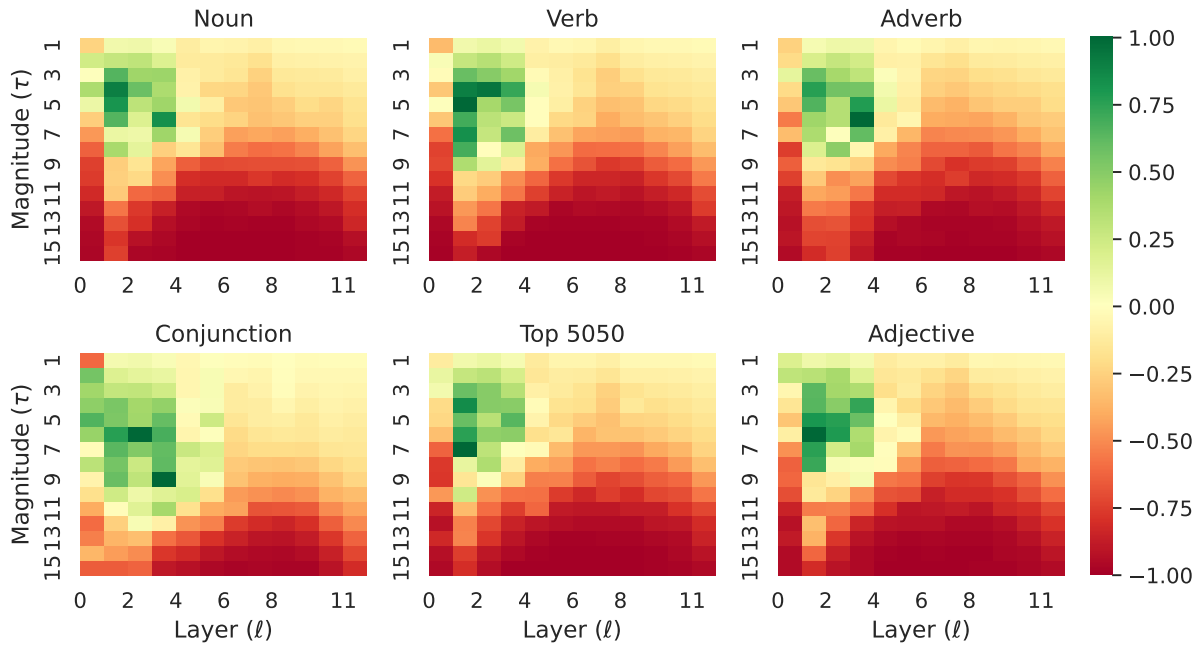

Figure 9: **GPT2-Small, *Hand* dataset.** Heatmap shows average percent difference between pre- and post-injection answer probabilities for multi-hop prompts excluding outliers not within ± 2 standard deviations from the mean across various parts of speech.
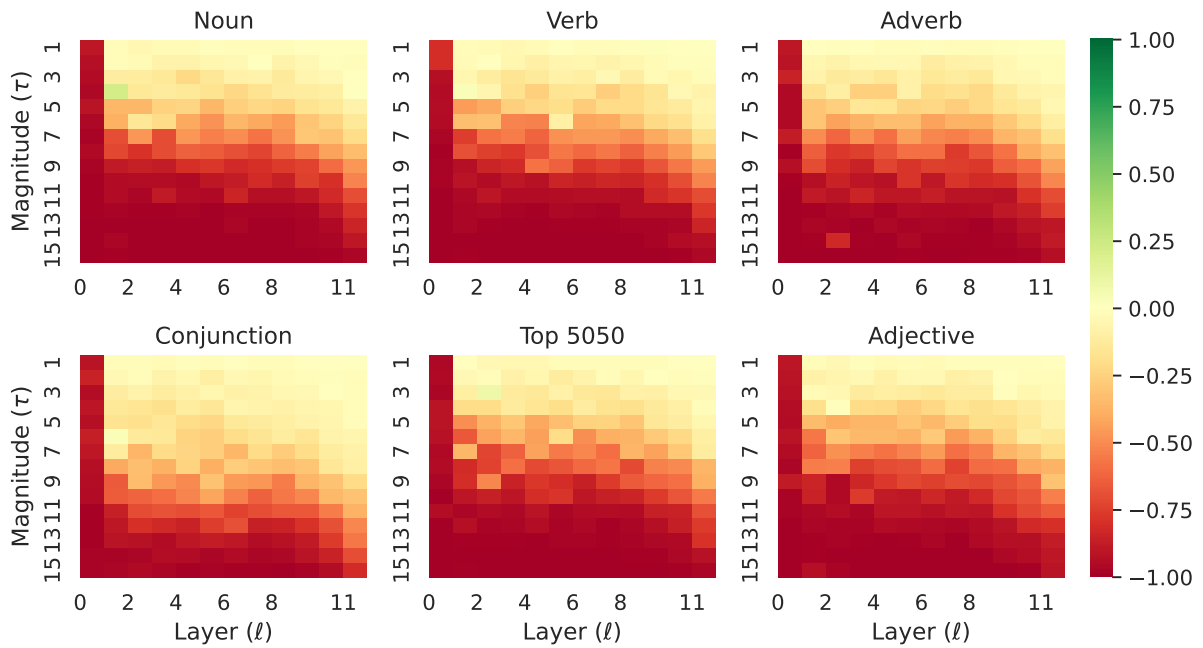
B Dataset Example Appendix

Dataset	Single-Hop Prompt	Multi-Hop Prompt
Hand	George Washington fought in the ...[Revolutionary War]	The first president of the United States fought in the ...[Revolutionary War]
	Burj Khalifa is located in the city of ...[Dubai]	The tallest building in the world is located in the city of ...[Dubai]
	Nelson Mandela brought an end to ...[Apartheid]	The first president of South Africa brought an end to ...[Apartheid]
	John F Kennedy was assassinated by a person named ...[Lee Harvey Oswald]	The 35th president of the United States was assassinated by a person named ...[Lee Harvey Oswald]
	The father of Hermes is ...[Zeus]	The father of the Greek messenger god is ...[Zeus]
2WMH	The place of birth of Dušan Hanák is ...[Bratislava]	The place of birth of the director of I Love, You Love is ...[Bratislava]
	The employer of Éric Rohmer is ...[Cahiers du cinéma]	The employer of the director of Triple Agent is ...[Cahiers du cinéma]
	The employer of Chip Gubera is ...[University of Missouri]	The employer of the director of Academy of Doom is ...[University of Missouri]
	Steve Vai received the ...[Grammy]	The performer of The Attitude Song received the ...[Grammy]
	The place of death of Augustus II the Strong is ...[Warsaw]	The place of death of the spouse of Christiane Eberhardine of Brandenburg-Bayreuth is ...[Warsaw]

Table 3: **Example prompts.** Single/multi-hop prompt pairs from *Hand* and *2WMH* datasets.

Multiple-Hop Prompt	Memory	Answer	Pre-injection Answer Prob.	Post-injection Answer Prob.
The God of Thunder is the son of ...	Thor	Odin	0.84%	3.37%
The first president to be assassinated succeeded in abolishing ...	Abraham Lincoln	slavery	30.46%	63.09%
The founder of Microsoft was born in the city of ...	Bill Gates	Seattle	1.55%	2.44%
The highest peak in the world is located in the ...	Mount Everest	Himalayan	3.40%	22.58%

Table 4: **Examples of memory injections.** Injecting memories with $\tau = 4, \ell = 9$ into GPT2-Small.

Prompt Type	Prompt	Layer ℓ	Head h	Output
Single-Hop	John F Kennedy was assassinated by a person named ...	10	0	[' Kennedy', ' JFK', ' Assass', ' assass', 'Kenn', ' assassination', ' Cuba', ' Oswald', ' assassin', ' Cuban', ' Fidel', ' Bobby', ' Havana', ' assassinated', ' assassins', ' Jackie', ' Castro', ' Jinn', ' assassinate', 'Mu', ' 1963', ' Kahn', ' drone', ' Cah', ' Mu', ' Ghosts', ' Soul', ' Laos', ' Cemetery', ' CIA']
	Barack Obama was a member of the ...	9	8	[' Obama', 'Obama', ' Maryland', ' America', ' JFK', ' Biden', ' Harlem', ' Washington', ' American', ' Clinton', ' White', ' Americans', ' Congressional', ' Harvard', ' Kennedy', ' FBI', ' Federal', ' CDC', ' DOJ', ' President', ' Georgetown', ' HHS', ' Barack', ' US', ' Trayvon', ' Connecticut', ' Holder', ' New', ' BLM', ' Baltimore']
	Cain murdered a person named ...	2	1	[' police', ',', ' the', ' a', '\n', ' and', ' violence', '.', ' death', ' in', ' criminal', ' of', ' to', ' victim', ' "', ' -', ' at', ' victims', ' crime', ' from', ' an', ' that', ' murder', ' crimes', ' is', ' was', ' he', ' for', ' (', ' killed']
	Russia is mostly located on the continent of ...	9	8	[' Moscow', ' Russian', 'Moscow', ' Russia', ' Kremlin', ' Putin', 'Putin', 'Russia', ' Russians', 'Russian', '◆', ' ◆', ' Dmitry', ' Mikhail', ' Vladimir', ' Sergei', ' Siberia', ' Soviet', ' Siberian', ' Ukraine', ' Ukrainian', ' Sochi', ' Caucasus', ' Nikol', 'Soviet', ' KGB', ' Dmit', ' USSR', 'Ukraine', ' Ukrainians']
	George Washington fought in the ...	9	8	[' Washington', 'Washington', ' Virginia', 'Virginia', ' Maryland', ' Congressional', ' Georgetown', ' Dull', ' Smithsonian', ' Maine', ' Burr', ' Jefferson', ' Navy', ' Capitol', ' congressional', ' FDR', ' Lexington', ' Byrd', ' Rhode', ' Roosevelt', ' Pike', ' Everett', ' Brookings', ' Madison', 'apeake', ' Randolph', ' VA', ' Arlington', ' Americans', ' Lafayette']
Multi-Hop	The 35th president of the United States was assassinated by a person named ...	10	0	[' assass', ' Assass', ' assassination', ' assassin', ' assassins', ' assassinate', ' Malik', ' bullets', ' gunmen', ' assassinated', 'Mu', ' Pakistani', ' sniper', ' killings', ' JFK', ' Pakistan', ' homicides', ' Alger', ' lethal', ' Islamabad', ' Karachi', ' shooting', ' gun', ' gunshot', ' Mu', ' murder', ' killing', ' pistols', ' murders', ' gunned']
	The first black president of the United States was a member of the ...	9	8	[' Negro', ' NAACP', ' blacks', ' black', ' Baltimore', ' White', ' negro', ' Washington', ' BLM', ' white', ' FBI', ' America', ' Maryland', ' African', ' Trump', ' Nixon', ' Charleston', ' Americ', ' KKK', 'Washington', ' Virginia', ' racial', ' Blacks', 'white', 'White', ' nig', ' Black', ' Obama', ' Louisiana', ' whites']
	Adam and Eve's eldest son murdered a person named ...	2	1	[',', ' the', ' and', ' a', ' "', ' in', '\n', '.', ' to', ' of', ' at', ' is', ' he', ' -', ' that', ' was', ' for', ' police', ' from', ' on', " '", ' as', ' death', ' had', " '", ' an', ' his', "'s", ' said', ' told']
	The largest country in the world is mostly located on the continent of ...	9	8	[',', '\n', ' the', ' and', '.', ' in', ' a', ' to', ' of', ' (', ' -', ' for', ' that', ' "', ' :', ' is', ' or', ' at', ' as', ' I', ' on', ' with', ' it', ' an', ' from', ' all', ' by', ' not', "'s", ' more']
	The first president of the United States fought in the ...	9	8	[' Trump', ' Washington', ' America', 'Washington', ' American', 'Trump', 'America', ' Obama', ' Donald', ' FBI', ' Congressional', ' Americans', 'American', ' Nixon', ' Congress', ' congressional', ' White', ' Roosevelt', ' Republican', ' Negro', ' Clinton', ' JFK', ' Reagan', ' Virginia', ' FDR', 'Obama', 'Americans', ' Americ', 'FBI', 'Congress']

Table 5: **Example of attention head outputs** from GPT2-Small for *Hand*.