Promote, Suppress, Iterate: How Language Models Answer One-to-Many Factual Queries

Tianyi Lorena Yan and **Robin Jia**
University of Southern California
{tianyi.yan, robinjia}@usc.edu

Abstract

To answer one-to-many factual queries (e.g., listing cities of a country), a language model (LM) must simultaneously recall knowledge and avoid repeating previous answers. How are these two subtasks implemented and integrated internally? Across multiple datasets, models, and prompt templates, we identify a promote-then-suppress mechanism: the model first recalls all answers, and then suppresses previously generated ones. Specifically, LMs use both the subject and previous answer tokens to perform knowledge recall, with attention propagating subject information and MLPs promoting the answers. Then, attention attends to and suppresses previous answer tokens, while MLPs amplify the suppression signal. Our mechanism is corroborated by extensive experimental evidence: in addition to using early decoding and causal tracing, we analyze how components use different tokens by introducing both *Token Lens*, which decodes aggregated attention updates from specified tokens, and a knockout method that analyzes changes in MLP outputs after removing attention to specified tokens. Overall, we provide new insights into how LMs' internal components interact with different input tokens to support complex factual recall. [1]

1 Introduction

Transformer-based language models (LMs) store a vast amount of factual knowledge in their parameters (Petroni et al., 2019; Dai et al., 2021; Geva et al., 2022). Many recent works have studied where and how LMs recall this knowledge for one-to-one factual queries, which ask the model to recall a single fact (e.g., the capital of a country) given a subject-relation pair (Meng et al., 2022; Geva et al., 2023; Merullo et al., 2023b).

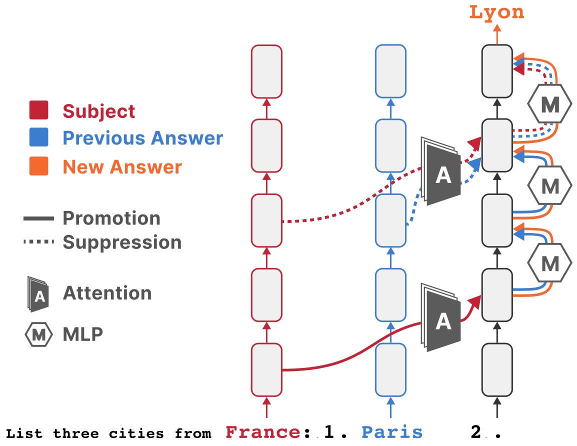

Figure 1: To answer one-to-many factual queries, we found that LMs first use attention to propagate subject information to the last token, which is used by MLPs to promote all possible answers. Attention then attends to and suppresses the subject and previous answer tokens, while MLPs amplify the suppression and further promote new answers.

In this work, we study the comparatively unexplored task of one-to-many knowledge recall (1MKR), in which the model must generate a list of answers without repetition. Many real-world relations, such as a country's cities or an artist's songs, are one-to-many. This more complex task requires LMs to integrate multiple pieces of contextual information, including the subject and previously generated answers, to simultaneously perform two subtasks: **knowledge recall** and **repetition avoidance**. We uncover LMs' mechanism for 1MKR by understanding (1) the overall process by which they generate distinct answers at different steps, and (2) how they perform both answer promotion and repetition avoidance.

To understand the overall process, we early decode (Nostalgebraist, 2020) the output of attention and MLPs to examine how the logits of the subject and answer tokens change across layers. We find that LMs first promote all answers and then suppress the ones that have been previously generated.

[1] Code is available at https://github.com/Lorenayannnnn/how-lms-answer-one-to-many-factual-queries.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 16111–16134
November 4-9, 2025 ©2025 Association for Computational Linguistics

Specifically, attention copies the subject information at the middle layers and MLPs promote all possible answers. Then, both components suppress previous answer tokens at late layers. These observations hold for both Llama-3-8B-Instruct and Mistral-7B-Instruct-v0.2 across three datasets.

To examine how LMs implement knowledge recall and repetition avoidance, we first run causal tracing (Meng et al., 2022) to locate tokens that are critical to LMs' outputs; these important tokens include the subject, previous answers, and the last token. Then, we analyze how both attention and MLP layers use these tokens. For attention, we propose Token Lens, a new technique that aggregates and then unembeds the results of attending to a given token or span; in this way, we can observe how attention to each token promotes or suppresses different output tokens. For MLPs, we design an attention knockout method inspired by Geva et al. (2023): we knock out the attention from the last token to the target tokens and examine the resulting change in MLP output logits to determine how MLPs use target token information. We find that LMs use both the subject and previous answer tokens for knowledge recall: attention propagates the subject information from the subject to the last token, and MLPs leverage the information and previous answer tokens to promote answers. In addition, previous answer tokens trigger suppression of themselves: attention attends to and suppresses previous answer tokens, while MLPs amplify the suppression signal. LMs aggregate this information at the last token to generate distinct answers across steps.

Overall, our study elucidates how LMs use attention and MLPs to interact with different tokens and perform knowledge recall and repetition avoidance for 1MKR. We hope this work opens pathways for analyzing more complex tasks requiring dynamic integration of contextual information.

2 Related Work

Interpretability of Language Models. Works on mechanistic interpretability aim to reveal the function of different components in LMs (Elhage et al., 2021; Bansal et al., 2022), such as neurons (Dai et al., 2021; Gurnee et al., 2023), attention heads (Michel et al., 2019; Olsson et al., 2022), and MLPs (Geva et al., 2020, 2022). In particular, how LMs store and use knowledge has been widely studied by many prior works (Petroni et al., 2019; Bouraoui et al., 2020; Cao et al., 2021; Dalvi et al., 2022; Da et al., 2021). However, most prior studies have mainly focused on one-to-one knowledge recall, where LMs retrieve a single fact given a subject-relation pair. In this work, we study how LMs' components contribute to one-to-many knowledge recall, which is a more complex setting that requires LMs to integrate multiple types of contextual information: subject, relation, and previously generated answers.

Attribution Methods. Prior works have introduced various methods for analyzing the function of different components, including probing (Burns et al., 2022; Li et al., 2024), patching (Goldowsky-Dill et al., 2023; Ghandeharioun et al., 2024), early decoding (Nostalgebraist, 2020; Merullo et al., 2023b), and knocking out component outputs to assess their impact on models' outputs (Chang et al., 2023; Li et al., 2023; Geva et al., 2023). Our method, Token Lens and attention knockout (inspired by Geva et al. (2023)) examines the importance of attention and MLPs by early decoding their token-level outputs, revealing how LMs use the two components to integrate information from various parts of the input.

Dissecting Component Functions. Recent works have studied the functions of MLPs and attention in knowledge recall given subject-relation pairs. Meng et al. (2022) and Geva et al. (2023) demonstrate that MLPs enrich subject representations at early layers, while Geva et al. (2022) and Merullo et al. (2023b) highlight how MLPs promote correct answer tokens by writing updates to the residual stream and adjusting the vocabulary probabilities. This mechanism is still essential for the model to generate multiple answers tied to the given subject. Prior works have also shown that attention and MLPs play a key role in extracting important tokens and suppressing repeated ones (Wang et al., 2022; McDougall et al., 2023; Voita et al., 2023; Merullo et al., 2023a; Tigges et al., 2024), which is essential for preventing the model from generating duplicate answers. Merullo et al. (2024) further decomposes attention heads and identifies low-rank subspaces in which components communicate to selectively inhibit repetitive items from a list given in the context, which also involves list processing and repetition avoidance but not recalling factual knowledge from model parameters.

3 Problem Settings

We first introduce the task of one-to-many knowledge recall and describe our experiment settings.

3.1 Task: One-to-Many Knowledge Recall

In 1MKR, a language model is given a subject entity s and a relation r, and must generate a set of corresponding object entities $O = \{o^{(1)}, o^{(2)}, \ldots, o^{(n)}\}$ that are related to s through r. All generated object entities must be distinct, that is, $o^{(i)} \neq o^{(j)}$ for $i \neq j$. For example, given $s =$ "U.S.A." and $r =$ "cities of", one possible valid set of object entities is $O = \{\text{Los Angeles}, \text{San Francisco}, \text{Seattle}\}$. To perform this task, the model must perform two key subtasks:

1. **Knowledge recall**: The model must identify and extract the subject s from the input and retrieve entities that are connected to s through the relation r from its internal knowledge.
2. **Repetition avoidance**: The model must not generate duplicate entities.

Possible mechanisms. Multiple different mechanisms could be used by the model to perform 1MKR. On one hand, the model could use different attention heads to promote a different answer at each timestep. It could first use suppression heads (Wang et al., 2022) to identify previously generated answers, then change the attention patterns of subsequent heads to avoid promoting those answers. Such a mechanism would mirror the use of suppression heads to avoid generating incorrect, repetitive tokens in the IOI task (Wang et al., 2022). To promote answers, the model could attend to the subject token position, which could encode different answers in different attention value vectors due to subject enrichment (Geva et al., 2023).

On the other hand, the model could first promote all relevant answers and then suppress previously generated ones. It could extract all possible answers from the subject representation (Geva et al., 2023; Meng et al., 2022), regardless of which object entities have been generated. Then, copy suppression heads could identify previous answer tokens and prevent the model from generating them, similar to McDougall et al. (2023). The results of knowledge recall and repetition avoidance could be additively combined in the residual stream to yield a correct and non-duplicate output, similar to Chughtai et al. (2024). In this paper, we uncover the true mechanism that the model uses for one-to-many knowledge recall.

3.2 Datasets and Models

We curate three 1MKR datasets on different topics: (1) cities of a country,[2] (2) songs performed by an artist,[3], and (3) movies acted in by an actor or actress.[4] A summary of the datasets is provided in Tab. 1. For each dataset, the number of object entities $n = 3$.[5] We filter out subjects that are associated with fewer than three object entities for the specified relation.

We study two LMs: Llama-3-8B-Instruct (AI@Meta, 2024) and Mistral-7B-Instruct-v0.2 (AI, 2024). We have three prompt templates for each model and dataset, which are shown in Appx. §A. To create the data for analyzing LMs' behaviors, we first generate three answers using greedy decoding, ensuring consistent outputs for examining component behaviors across different answer steps. We then retain the entries where all three predicted answers are correct to focus on cases where the models' knowledge is accurate.

Tab. 1 shows the number of correct predictions made by the models across the datasets. The low accuracy may be explained by (1) long-tail entities (e.g., less popular actors or songs), (2) outdated datasets compared to the model's knowledge, and (3) the strict use of exact match evaluation (e.g., "Mission: Impossible" is considered incorrect even if given "Mission: Impossible - Fallout" is in the label list). For all (dataset, model) pairs, we have at least 100 correct instances, providing a sufficient sample size for the analysis. For the rest of the paper, we focus only on the correct cases. When analyzing models' behaviors at step i ($i = 1, 2, 3$), we keep all tokens before the first token of the ith answer as input. Refer to Appx. §B for examples and details. We report results macro-averaged across all models, datasets, and prompt templates in the main section. Refer to the appendix for full results of all answer steps and specific models and datasets. We run all experiments on a single RTX A6000 GPU.

[2] https://simplemaps.com/data/world-cities

[3] https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube

[4] https://www.kaggle.com/datasets/darinhawley/imdb-films-by-actor-for-10k-actors

[5] We also tested larger n. The models fail to generate at least 100 correct cases except for the Actor-Movies dataset with $n = 5$, where all major results align with those discussed in the main section. See Appx. §F for details.

Dataset	Subject (s)	Relation (r)	Object (o)	# Entries	Llama-3-8B-Instruct (Acc)	Mistral-7B-Instruct-v0.2 (Acc)
Country-Cities	Country	contains	Cities	168	122/168 (72.8%)	118/168 (70.2%)
Artist-Songs	Artist	performer of	Songs	2077	276/2077 (13.3%)	221/2077 (10.6%)
Actor-Movies	Actor	acted in	Movies	8790	1235/8790 (14.1%)	799/8790 (9.1%)

Table 1: Models' performance on all three datasets averaged across three prompt templates. Lower accuracy may be attributed to long-tail entities, outdated data, and overly strict exact-match evaluations. Our analysis focuses on the correct cases. See Appx. §A for the prompt templates and per-template performance.

4 Decoding the Overall Mechanism

To understand how LMs perform 1MKR, we first inspect the outputs of attention and MLP across layers. We aim to understand how knowledge recall and copy suppression coordinate to produce different correct answers across generation steps.

4.1 Method: Decoding Component Outputs

Given a transformer LM with L layers, each layer l has a multi-headed attention (MHA) and a MLP layer for $l = 1 \ldots L$. Let $a^{(l)} \in \mathbb{R}^d$ and $m^{(l)} \in \mathbb{R}^d$ be the outputs of the MHA and MLP at layer l at the last token position[6] respectively. Similar to Nostalgebraist (2020) and Geva et al. (2022), we (early) decode $a^{(l)}$ and $m^{(l)}$ by passing them through the final layer layernorm and unembedding matrix $U \in \mathbb{R}^{|\text{Vocab}| \times d}$ and obtain the logits to examine their contributions to knowledge recall and repetition avoidance:

$$\text{logits} = U \cdot \text{LayerNorm}(z^{(l)}) \qquad (1)$$

where $z^{(l)}$ is $a^{(l)}$ or $m^{(l)}$, and $\text{LayerNorm}(\cdot)$ denotes the final layernorm. In this paper, $\text{LayerNorm}(\cdot)$ is the RMSNorm (Zhang and Sennrich, 2019). Note that the RMSNorm is calculated based on the input's hidden state from the final layer, not directly on $a^{(l)}$, ensuring consistent normalization across layers and components (Chang et al., 2024).

4.2 LMs Promote Then Suppress

We analyze the logit values of the first tokens of object entities predicted across three answer steps and the subject. A positive logit indicates promotion, while a negative logit suggests suppression. Our analysis shows that LMs use both attention and MLPs to promote all possible answers at each step while suppressing repetitions.

Attention primarily copies subject information. As shown in Fig. 2, attention outputs positive logits for the subject token in the middle layers across all three answer steps. While the three answers are slightly promoted at layer 25, their logits are still close to zero and have a smaller magnitude compared to that of the subject at the middle layers. This pattern indicates that attention copies or propagates subject information at the last token position. Interestingly, the answer promotion pattern is more evident in the Country-Cities dataset (Fig. 10, Fig. 11) but not in the Artist-Songs and the Actor-Movies datasets (Fig. 12, Fig. 13, Fig. 14, Fig. 15).

MLPs promote all possible answers. From the middle to later layers, MLPs consistently output positive logits for all three possible answers (Fig. 2). These logits increase across generation steps, with their magnitude significantly exceeding that of attention logits. These findings suggest that MLPs strongly promote all possible answers regardless of prior predictions, thereby providing a stronger answer promotion signal compared to attention.

Previously generated answers are suppressed at later layers. Both attention and MLPs suppress answers that have been generated previously. Starting from layer 28, attention outputs negative logits for $o^{(1)}$ at step 2 and for both $o^{(1)}$ and $o^{(2)}$ at step 3 (Fig. 2). Similarly, MLPs decrease the logit of previous answers at the same layer. Since MLPs themselves cannot attend back to early tokens, this suppression likely results from leveraging suppression signals from attention, a hypothesis further investigated in §6.3.

In the final layers, both attention and MLPs increase answers' logits, especially those that have not been generated. This pattern may be explained by how LMs use the final layers to adjust the logits and regulate the confidence or certainty of their predictions (Stolfo et al., 2024). Overall, all the observations above demonstrate that LMs promote

[6] We focus on the last token position as the model directly uses it to generate the next answer.

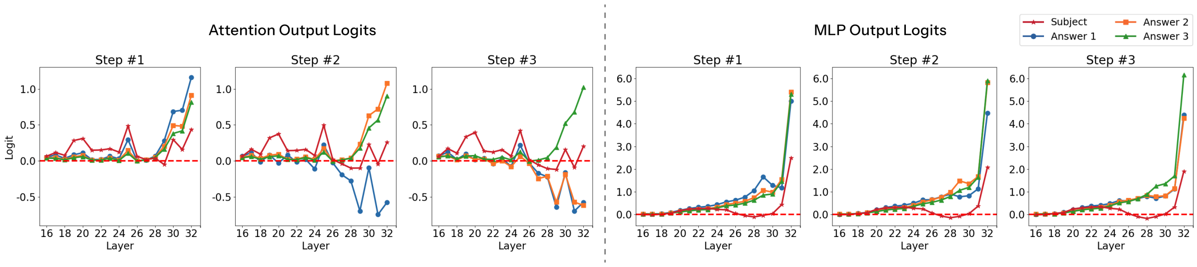

Figure 2: Logit of the subject and answer tokens from unembedding attention and MLP outputs. Attention primarily promotes the subject at the middle layers, then promotes new answers and suppresses previous answers at deeper layers. MLPs consistently promote all answers; at deeper layers, they also decrease the logits of previously generated answers. Early layers are omitted as logits are near zero. See Appx. §C for full figures.

all three answers and then suppress previously generated ones.

5 Which Tokens Matter?

To better understand the promote-then-suppress mechanism, we now investigate how LMs implement knowledge recall and repetition avoidance. In this section, we use causal tracing (Meng et al., 2022) to identify the input tokens that most influence model predictions. In §6, we analyze how these tokens are used by attention and MLPs to facilitate knowledge recall and repetition avoidance.

5.1 Which Tokens Should Be Noised?

Prior work shows that in order to recall knowledge, LMs encode information about relevant object entities in subject tokens and retrieve this information via attention (Geva et al., 2023; Meng et al., 2022). Other work shows that LMs avoid repetition by using attention heads to attend to previous tokens and suppress them (McDougall et al., 2023; Wang et al., 2022; Merullo et al., 2023a). Thus, we hypothesize that the subject and previous answer tokens play decisive roles in our two key sub-tasks (§3.1).

To confirm these hypotheses, we use causal tracing (Meng et al., 2022): we separately add noise to the subject and previous answer tokens, restore selected components' activations to their values without noise, and visualize the difference in the probability of $o^{(i)}$ that will be predicted at each answer step i before and after the restoration. This approach allows us to measure the impact of specific token activations on the models' outputs.

Intervention on Subject. Fig. 3 visualizes the impact of attention and MLPs on LMs' predictions when intervening on the subject tokens at step 2 (Refer to Appx. §D for figures of other answer steps and specific models and datasets, which have similar patterns). The probability difference peaks around or above 0.55 for both components, confirming our hypothesis that the subject plays a crucial role in knowledge recall. Attention's contributions peak in the middle layers at the last token, while MLPs dominate in early layers at the subject token and in late layers at the last token. These observations suggest that attention propagates subject information from early MLP layers to the last token, where MLPs may leverage it for answer promotion, as discussed in §6.2.

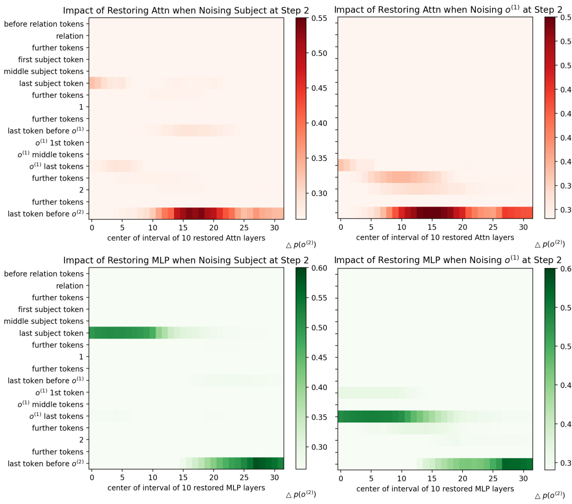

Figure 3: The impact of attention and MLPs' activations on LMs' predictions when intervening on the subject (left) and previous answer tokens (right) at step 2. The probability differences all peak around or above 0.55, reflecting the importance of both the subject and previous answer tokens. See Appx. §D for figures of other answer steps, which have similar patterns.

Intervention on previous answers. Noising previous answer tokens also leads to high probability changes in LMs' output probabilities, with an average difference of around or above 0.55 across answer steps (Fig. 3). This finding supports our hypothesis that previous answer tokens are also critical to LMs' outputs. Similar to the results of noising the subject, attention's contributions peak in both the middle and the last layers at the last token. MLPs dominate in early layers at the previous answer positions and in late layers at the last token, reflecting that the previous answer tokens are used by both components to make nontrivial contributions to models' predictions.

6 Analyze Critical Tokens

The causal tracing analysis confirms that both the subject and previous answer tokens are important for handling one-to-many factual queries. To determine whether the subject primarily supports knowledge recall and previous answer tokens drive suppression, we next analyze how attention and MLPs utilize these tokens, as well as the last token that the model uses to predict the next answer.

6.1 Methodology for Analyzing Tokens

To analyze how attention and MLPs utilize the subject, previous answer, and last tokens, we develop techniques to unembed their token-specific outputs and examine their roles in knowledge recall and suppression.

Attention: Token Lens. For attention, we propose Token Lens, a new technique that unembeds the aggregated outputs of attention to specified tokens. Let $t = \{t_1, ...t_k\}$ denote the target tokens we are examining. t can be the subject s, an object entity answer $o^{(i)}$, or the last token of the input. Let $a^{(l_i)}$ be the ith attention head in layer l of a transformer LM, for $i = 1 \dots n$ and $l = 1 \dots L$. Let $p_{t_j}^{(l_i)} \in \mathbb{R}$ denotes $a^{(l_i)}$'s attention weight between the last input token[7] and the t_jth token of the input. Similarly, let $v_{t_j}^{(l_i)} \in \mathbb{R}^{d_{\text{head}}}$ denotes the value vector of $a^{(l_i)}$ for the the t_jth token.

We first gather the information that each attention head $a^{(l_i)}$ aggregates from all target tokens, which is calculated as the sum of all weighted value vectors of t of $a^{(l_i)}$:

$$a_e^{(l_i)} = \sum_{j=1}^{k} p_{t_j}^{(l_i)} \cdot v_{t_j}^{(l_i)} \tag{2}$$

Then, the full attention output of the target tokens from the lth layer is:

$$a_e^{(l)} = W_o^{(l)} \cdot \text{Concat}(a_e^{(l_1)}, \dots, a_e^{(l_n)}) \tag{3}$$

where $W_o^{(l)} \in \mathbb{R}^{d \times n d_{\text{head}}}$ is the output projection matrix of layer l. This vector $a_e^{(l)} \in \mathbb{R}^d$ represents the contribution of MHA at layer l to the output from the target tokens.

Finally, following the same approach of (early) decoding attention and MLP outputs in §4.1, we unembed $a_e^{(l)}$ to obtain the logits of the first token of the subject and answers and examine how attention uses the target tokens to perform promotion or suppression.

MLPs: Attention Knockout. Since MLPs themselves cannot attend to previous tokens–a function exclusive to MHA–we adopt an attention knockout approach inspired by Geva et al. (2023). By knocking out the attention from the last token to the target tokens, we examine changes in MLP output logits to determine how MLPs utilize target token information for knowledge recall and repetition avoidance. Specifically, we zero out the attention weights between the last and the target tokens:

$$p_{t_j}^{(l_i)} \leftarrow 0, \forall i \in [1, n], \forall j \in [1, k], \forall l \in [1, L]$$

Let $m^{(l)}$ and $m'^{(l)}$ denote the MLP output at layer l before and after applying the attention knockout respectively. We unembed these outputs using the same early decoding approach described in §4.1. By subtracting the logits derived from $m'^{(l)}$ from those of $m^{(l)}$, we examine the difference in the logits of the subject and the answer tokens. A positive difference value indicates MLPs use the knocked-out tokens to promote a token; a negative difference means suppression.

6.2 Role of Subject Tokens

Across all models and datasets, attention and MLPs use subject tokens to contribute to answer promotions while suppressing the subject itself.

Attention first moves the subject to the last token position. As shown in Fig. 4, attention to the subject greatly increases the subject token's logit at the middle layers. To a lesser degree, it also promotes answer tokens, particularly at layer 25.[8] Answer promotion is most pronounced in the Country-Cities dataset (Fig. 24, Fig. 25) but less evident in the Artist-Songs and Actor-Movies datasets (Fig. 26, Fig. 27, Fig. 28, Fig. 29). In all datasets, the subject logit is still larger than that of

[7] We only need to do the analysis when LLMs start to generate the next answer. Therefore, we are only looking at the last token of the input.

[8] Thus, the observation from §4.2 that attention promotes answers at layer 25 can be attributed to the subject token.

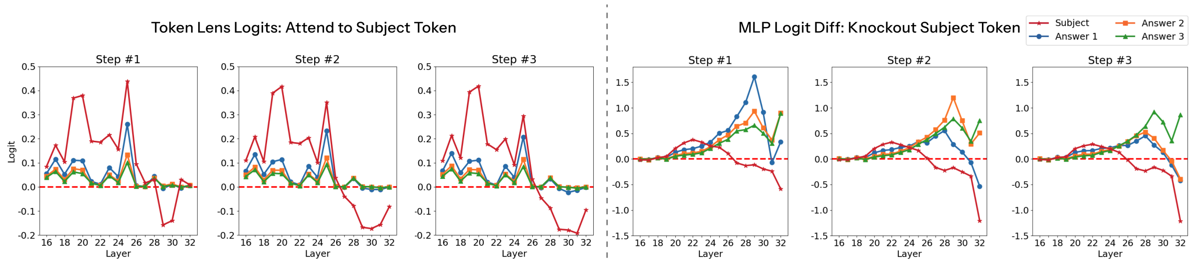

Figure 4: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens when attending to or knocking out the subject tokens. Attention promotes and extracts subject information in the middle layers but suppresses it in later layers. MLPs promote the answers and suppress the subject at deeper layers. Refer to Appx. §E for full figures.

each answer across all answer steps, demonstrating that attention primarily copies or propagates subject information from the subject to the last token position.

MLPs use the subject to promote answers. From the middle to late layers, MLP logit differences of the answer tokens are all positive across answer steps. Combined with attention's promotion of the subject, our findings suggest a coordinated mechanism: attention propagates subject information to the last token, and MLPs leverage this information to promote relevant answers.

At late layers, attention shifts from promoting to suppressing the subject. Starting around the 28th layer, attention outputs negative logits for the subject tokens. This transition shows that while attention initially promotes the subject, it later suppresses the subject to prevent incorrect generations, as the subject itself is not a correct answer.

MLPs amplify subject suppression. The MLPs' logit differences for the subject token become negative in later layers, especially at steps 2 and 3. This pattern illustrates that MLPs not only promote answers but also actively suppress the subject when it is no longer relevant for the next prediction. Combined with attention's suppression of the subject at later stages, our result suggests that MLPs amplify suppression signals from attention to prevent incorrect generations.

6.3 Role of Previous Answer Tokens

Attention plays a crucial role in suppressing repetitions. Attention consistently outputs negative logits for previous answer tokens at both step 2 and step 3 in the final layers. This result shows that attention attends to and suppresses tokens that have already appeared in the context, ensuring previously generated answers are not repeated.

MLPs amplify suppression of previous answers. As shown in Fig. 5, all previous answer tokens have negative MLP logit differences at late layers. For instance, o_1 has negative logits at step 2 starting around layer 27; o_1 and o_2 exhibit similar patterns at step 3. This suppression aligns with attention's role in inhibiting previously generated tokens, suggesting that MLPs amplify these suppression signals to prevent repetition.

MLPs also use previous answer tokens for knowledge recall. Surprisingly, we observe positive MLP logit differences for new answers across answer steps (Fig. 5). Specifically, the logit differences of both o_2 and o_3 are positive when intervening on o_1 at step 2; o_3 has positive logits differences when intervening on o_1 or o_2 at step 3. This pattern shows that MLPs also leverage previous answer tokens to promote new answers. Since LMs already promote all relevant answers when predicting previous answers, it is plausible that the models reuse these prior computations to promote new answers. These findings show that the subject token is not the sole source of answer promotion (§5.1). The previous answer tokens also have a positive (but smaller) effect on answer promotion.

6.4 Role of Last Token

The last token aggregates knowledge recall and suppression information in the final layers to promote all answers while prioritizing the correct answer for each step.

Attention promotes all answers at the last token in the final layers. Starting from layer 28, attention from the last token to itself significantly increases the logit of all three answers (Fig. 6). At

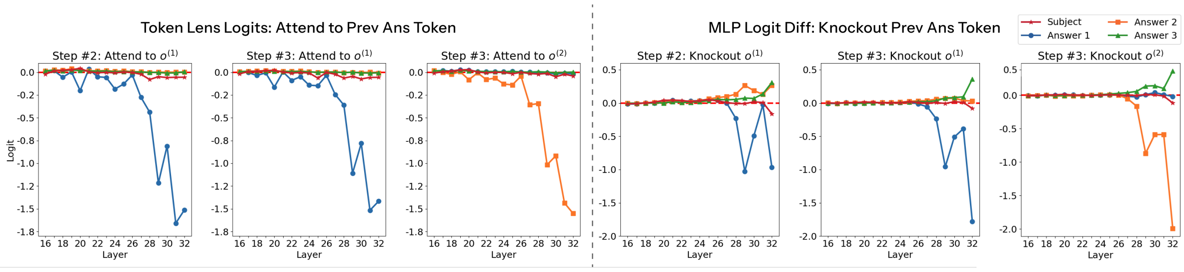

Figure 5: Token Lens logits of the attended previous answers (left) are negative at deeper layers, showing that attention suppresses prior answers. Negative MLP logit differences (right) for previous answers and positive differences for new answers suggest that MLPs use previous answer tokens for both repetition avoidance and knowledge recall.

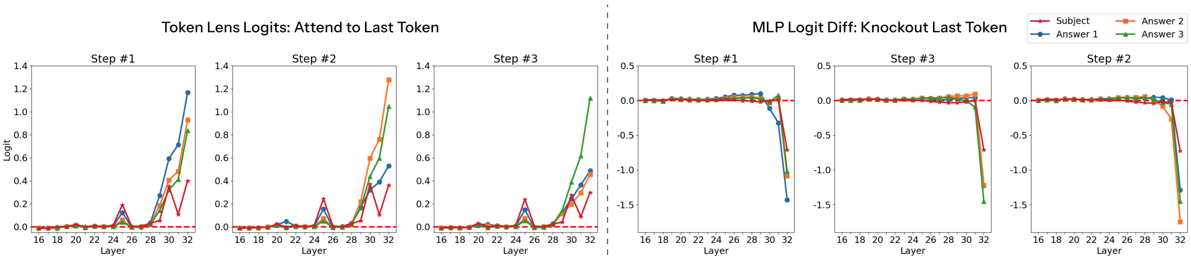

Figure 6: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens when attending to or knocking out the last token. Attention promotes all three answers and the subject at the final layers, prioritizing $o^{(i)}$ at each step i. Late-layer MLP logit differences are negative for the subject and answers, possibly compensating for the absence of direct attention to the last token to encourage correct outputs.

each step i, the logit for the answer $o^{(i)}$ is consistently the highest among the three answers. This result suggests that attention at the last token aggregates information from earlier layers related to knowledge recall and suppression, preparing the model for generating the next prediction.

MLPs compensate answer promotions when the direct attention to the last token is absent. Interestingly, we observe MLPs output negative logit differences for the subject and all three answers in the final layers when knocking out the attention from the last token to itself (Fig. 6). The answer $o^{(i)}$ for each step i consistently has the most negative logit differences. In other words, without having access to the attention output of the last token, MLPs output even higher logits for the subject and the answers. This behavior suggests a backup mechanism: without direct attention to the last token that aggregates information from early input tokens, the model may not have sufficient promotion and differentiation of the three answers. MLPs compensate this by further promoting the three answers to encourage the predictions to be correct. Similarly, Wang et al. (2022) find backup token mover attention heads that become active when the original token mover heads are ablated.

7 Are Knowledge Recall and Suppression Independent?

Observing that LMs promote all answers while suppressing previously generated ones, another question we have is whether knowledge recall and suppression are independent. To investigate this, we analyze the behavior of individual attention heads at the last token position to determine if they perform one, both, or neither of the two subtasks.

7.1 Methodology: Characterizing Attention Heads' Behavior

Our methodology involves the following steps:

1. Decode Attention Head Outputs: For each attention head, we decode its output at the last token position and collect the logits of the first token of t for a given input, where $t \in \{s, o^{(1)}, o^{(2)}, o^{(3)}\}$.

2. Calculate Layer-wise Baseline: For each layer l, we compute the mean μ_l and standard deviation σ_l of attention head logits across all heads in the layer.

3. Characterize Head Behavior: Let logit$(a_t^{(li)})$ denote the logit for the first token of t from attention head $a^{(li)}$. The behavior of $a^{(li)}$ on token t is classified as:

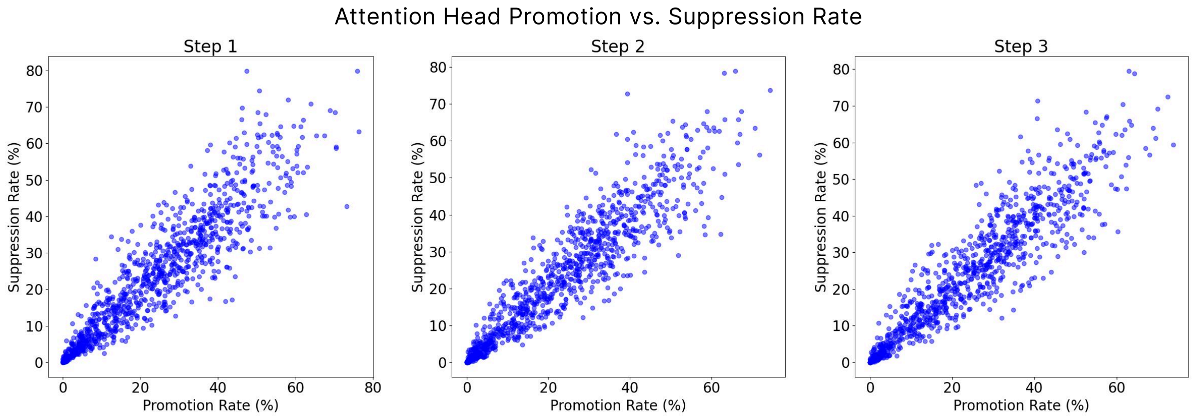

Figure 7: Promotion rate versus suppression rate of all attention heads across three answer steps macro-averaged across all models and datasets with template 1. The promotion rate and suppression rate positively correlate with each other, suggesting that answer promotion and suppression may not be independent of each other.

$$\text{Behavior}(a_t^{(li)}) = \begin{cases} \text{Promotion}, & \text{if } \text{logit}(a_t^{(li)}) > \mu_l + \sigma_l \\ \text{Suppression}, & \text{if } \text{logit}(a_t^{(li)}) < \mu_l - \sigma_l \\ \text{None}, & \text{otherwise} \end{cases}$$

4. Classify Head Function: $a^{(li)}$ is classified as performing promotion for the given input if it promotes the first token of any $t \in s, o^{(1)}, o^{(2)}, o^{(3)}$ and as performing suppression if it suppresses any such token.

5. Aggregate Results: We average the percentage of times each attention head is identified as performing promotion or suppression.

Then, by plotting the promotion rate against the suppression rate for all heads, we examine how LMs divide the labor among heads for knowledge recall and suppression.

7.2 Knowledge Recall and Suppression May Not be Independent

As can be observed in Fig. 7, the promotion rate and suppression rate of attention heads consistently correlate with each other across all three answers steps, with the majority of the data points concentrated in the bottom-left region of the plots. This finding shows that most attention heads contribute moderately to the two subtasks and are responsible for both token promotion and suppression, suggesting that knowledge recall and suppression may not be independent.

8 Conclusion

We uncover how language models answer one-to-many factual queries across two models and three datasets. By unembedding the output of attention and MLPs across layers, we find that LMs promote all answers and then suppress previously generated ones. We then delve into how LMs implement knowledge recall and repetition avoidance. We find that LMs use both the subject and previous answer tokens to perform knowledge recall. Attention first propagates subject information from the subject to the last token, which is then used by MLPs to promote all correct answers. At the same time, MLPs also utilize previous answer tokens to promote new answers at late layers. In addition, previous answer tokens trigger suppression of themselves. In the final layers, attention suppresses repetitions by attending to and outputting negative logits for previously generated answer tokens. MLPs reinforce and amplify this suppression by decreasing the logits of previous answer tokens around the same layers. At last, by integrating all relevant information for knowledge recall and suppression at the last token position, LMs effectively generate correct and distinct answers at different steps. We hope our findings encourage a deeper understanding of how LMs' internal components interact with context tokens to support complex factual recall and response generation.

Future Work. Future work could investigate possible redundancies in the model, as multiple tokens—such as the subject and previous answers—contribute to promoting new answers. This result raises the question of whether LMs redundantly encode knowledge and if it is necessary. Additionally, our analyses only focus on the correct cases. Examining the patterns when LMs use unreliable signals for factual recall or hallucinate could provide insights for mitigating such errors (Saynova et al., 2024).

Limitations

Our analyses primarily rely on Logit Lens (Nostalgebraist, 2020), which early decodes component outputs using LMs' last unembedding layer. While this method is training-free, it may be less reliable, particularly for early layers. More expressive techniques, such as Tuned Lens (Belrose et al., 2023) and SAE (Templeton et al., 2024), could be applied for a better understanding of 1MKR. Also, we use a single prompt template for each model and dataset. Further studies are needed to determine whether our findings generalize across different prompt templates.

While we attempt to identify how LMs recall knowledge, it is difficult to disentangle where the model truly recalls knowledge from its parameters, and where it amplifies already-recalled knowledge stored in the residual stream. This is especially difficult because models could redundantly encode knowledge in multiple places, and thus parametric recall and amplification could be interleaved. We hope future work can develop reliable methods for disentangling these concepts and lead to a more precise understanding of the underlying mechanism.

Acknowledgments

We thank Ting-Yun Chang for her helpful feedback on the paper. RJ was supported in part by the National Science Foundation under Grant No. IIS-2403436. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Mistral AI. 2024. Mistral 7b instruct model card.

AI@Meta. 2024. Llama 3 model card.

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. 2022. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. *arXiv preprint arXiv:2212.09095*.

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*.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. *arXiv preprint arXiv:2106.09231*.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2023. Do localization methods actually localize memorized data in llms? *arXiv preprint arXiv:2311.09060*.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2024. When parts are greater than sums: Individual llm components can outperform full models. *arXiv preprint arXiv:2406.13131*.

Bilal Chughtai, Alan Cooney, and Neel Nanda. 2024. Summing up the facts: Additive mechanisms behind factual recall in llms. *arXiv preprint arXiv:2402.07321*.

Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. 2021. Analyzing commonsense emergence in few-shot knowledge models. *arXiv preprint arXiv:2101.00297*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in bert. *arXiv preprint arXiv:2205.07237*.

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

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 Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*.

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

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Maximilian Li, Xander Davies, and Max Nadeau. 2023. Circuit breaking: Removing model behaviors with targeted ablation. *arXiv preprint arXiv:2309.05973*.

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

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

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

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023b. Language models implement simple word2vec-style vector arithmetic. *arXiv preprint arXiv:2305.16130*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Talking heads: Understanding inter-layer communication in transformer language models. *arXiv preprint arXiv:2406.09519*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Nostalgebraist. 2020. Interpreting gpt: The logit lens.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Denitsa Saynova, Lovisa Hagström, Moa Johansson, Richard Johansson, and Marco Kuhlmann. 2024. Fact recall, heuristics or pure guesswork? precise interpretations of language models for fact completion. *arXiv preprint arXiv:2410.14405*.

Alessandro Stolfo, Ben Wu, Wes Gurnee, Yonatan Belinkov, Xingyi Song, Mrinmaya Sachan, and Neel Nanda. 2024. Confidence regulation neurons in language models. *arXiv preprint arXiv:2406.16254*.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. Llm circuit analyses are consistent across training and scale. *arXiv preprint arXiv:2407.10827*.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*.

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

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

A Prompt Templates

Refer to Tab. 2 for the prompt templates that we use for each model and dataset. See Tab. 3 for number of correct cases from each model, dataset, and template.

B Sample Responses and Example of Analysis Data Creation

Following are some sample responses from the model: Llama-3-8B-Instruct:

- List three cities from China: 1. Beijing 2. Shanghai 3. Guangzhou
- List three songs performed by Ed Sheeran: 1. "Shape of You" 2. "Thinking Out Loud" 3. "Photograph"
- List three movies acted by Meryl Streep: 1. The Devil Wears Prada (2006) 2. The Iron Lady (2011) 3. Sophie's Choice

Mistral-7B-Instruct-v0.2:

- List the name of three cities from China:\n\n1. Beijing\n2. Shanghai\n3. Guangzhou
- List the name of three songs performed by Ed Sheeran: 1. Shape of You, 2. Perfect, 3. Thinking Out Loud
- List the name of three movies acted by Meryl Streep: 1. The Devil Wears Prada (2006)\n2. Sophie's Choice (1982)\n3. Kramer vs. Kramer

We filter out responses that contain incorrect object entities and only focus on the correct cases for analyses. For the artist-songs dataset, we use the Spotify API [9] to extend the song lists and keep them more up-to-date. To create data for analyzing, for example, Mistral-7B-Instruct-v0.2's behavior when predicting the first answer about Ed Sheeran, we will use "List the name of three songs performed by Ed Sheeran: 1." as the input and examine models' behavior when predicting "Shape".

C Decoding Attention and MLP Outputs Results

Fig. 8 and Fig. 9 are the full figures of logit of the subject and target entity tokens from decoding attention and mlp output across layers and answer steps. Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15 are the figures for specific models and datasets. As can be seen from Fig. 10 and Fig. 11, attention performing answer promotion at middle layers is more evident in the Country-Cities dataset. However, it is much less evident in the other two datasets (Fig. 12, Fig. 13, Fig. 14, Fig. 15). Refer to https://drive.google.com/drive/folders/1Xnk3lPLuqjmNABfrJvcJ4mSM9EBvYoub?dmr=1&ec=wgc-drive-globalnav-goto for the figures without early layers omitted.

D Causal Tracing Results

Fig. 16 and Fig. 17 are the full figures for causal tracing when noising the subject and previous answer tokens across all three answer steps and templates. Refer to https://drive.google.com/drive/folders/1aG-GZEIZ_EgUKQ8Vhe_Lv0mHILxIZfms?dmr=1&ec=wgc-drive-globalnav-goto for figures of specific models and datasets.

E Critical Token Analysis Results

Fig. 18, Fig. 19, Fig. 20, Fig. 21, Fig. 22, Fig. 23 are the complete results for Token Lens and Attention Knockout analyses on the subject token, previous answer tokens, and the last token. The results are macro-averaged across three answer steps and aggregated over all models and datasets. Fig. 24, Fig. 25, Fig. 26, Fig. 27, Fig. 28, Fig. 29 are the Token Lens and attention Knockout results on the subject token from different models and datasets. The pattern of attention using the subject token to promote answers is more prominent in the Country-Cities dataset (Fig. 24, Fig. 25) compared to the other two datasets (Fig. 26, Fig. 27, Fig. 28, Fig. 29). Refer to https://drive.google.com/drive/folders/1HtMtgm63ZZDfAnjeFDLJqMwSvLSyWAlj?dmr=1&ec=wgc-drive-globalnav-goto for dataset- and model-specific figures on all different tokens without early layers omitted.

F Analysis on More Answer Steps

F.1 Five Answer Steps

We asked the models to generate five object entities with prompt template 1. However, only the Actor-Movies dataset yielded over 100 correct cases from both models. The other datasets did not meet

[9] https://developer.spotify.com/documentation/web-api

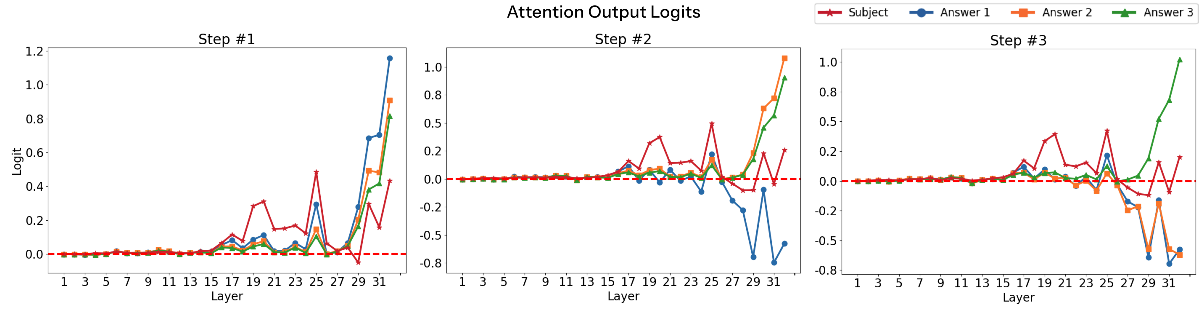

Figure 8: Logit of the subject and answer tokens from decoding the attention outputs across layers and answer steps. Attention primarily promotes the subject at the middle layers while promoting new answers and suppressing previously generated ones at deeper layers.

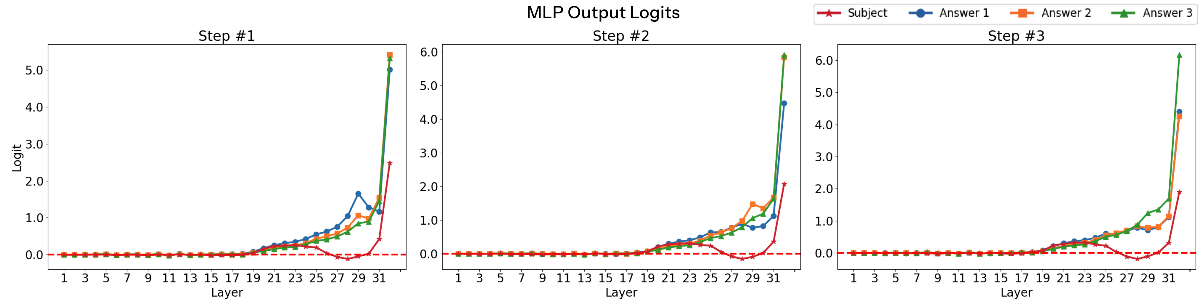

Figure 9: Logits of the subject and answer tokens from decoding the MLP outputs across layers and answer steps. The consistently positive logits for all three answers illustrate that MLPs promote multiple answers simultaneously. MLPs also decrease the logits of previously generated answers in deeper layers, contributing to repetition suppression alongside attention.

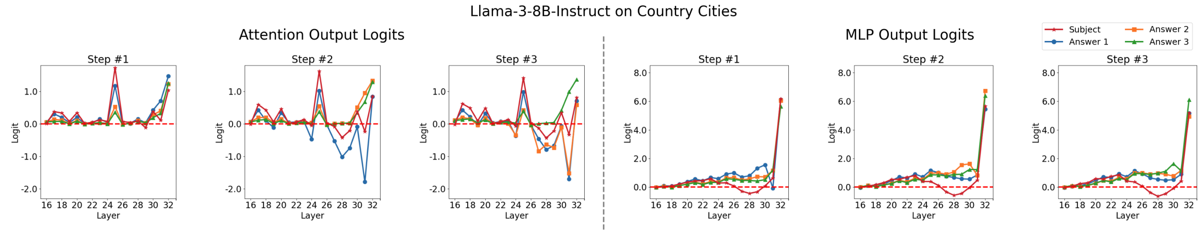

Figure 10: Attention and MLP output logits of Llama-3-8B-Instruct on Country-Cities dataset averaged across three prompt templates.

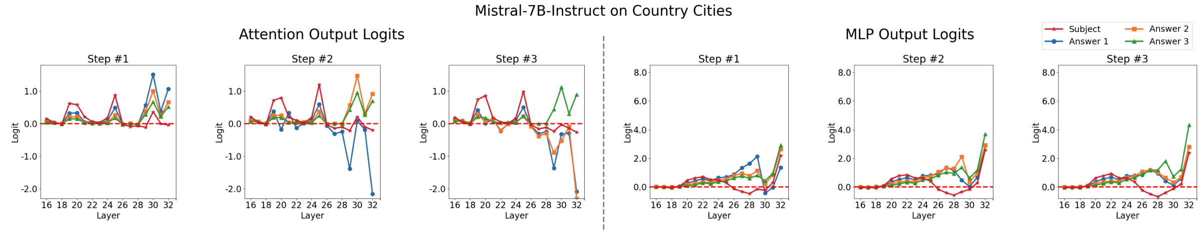

Figure 11: Attention and MLP output logits of Mistral-7B-Instruct on Country-Cities dataset averaged across three prompt templates.

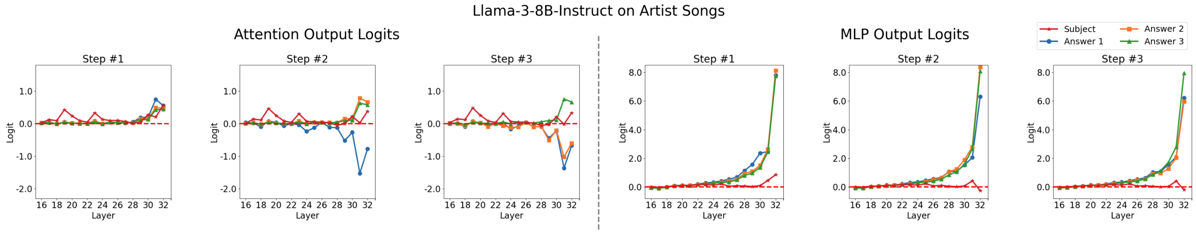

Figure 12: Attention and MLP output logits of Llama-3-8B-Instruct on Artist-Songs dataset averaged across three prompt templates.

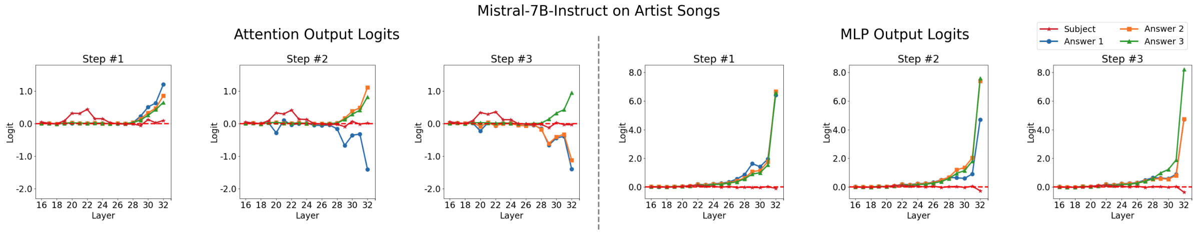

Figure 13: Attention and MLP output logits of Mistral-7B-Instruct on Artist-Songs dataset averaged across three prompt templates.

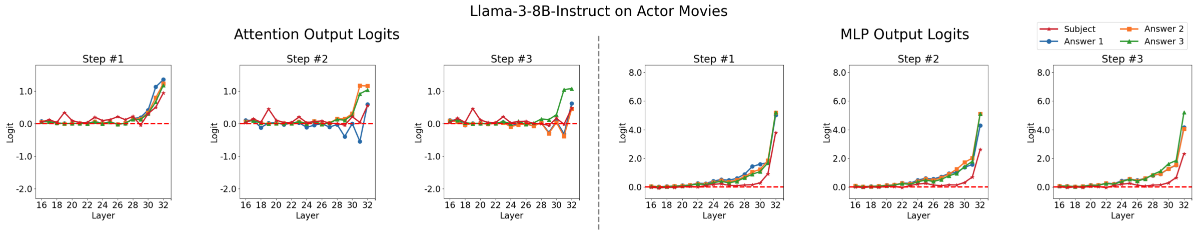

Figure 14: Attention and MLP output logits of Llama-3-8B-Instruct on Actor-Movies dataset averaged across three prompt templates.

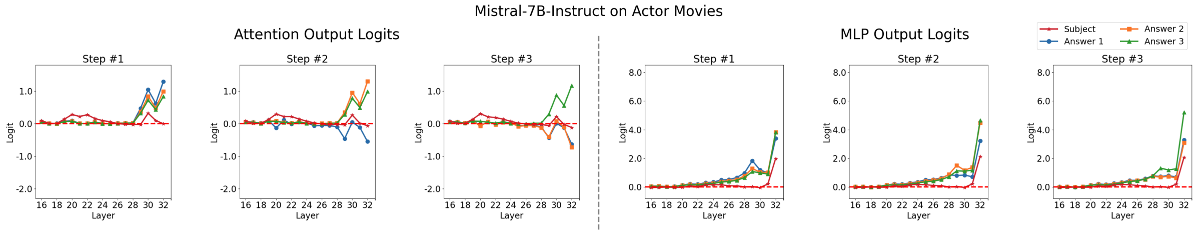

Figure 15: Attention and MLP output logits of Mistral-7B-Instruct on Actor-Movies dataset averaged across three prompt templates.

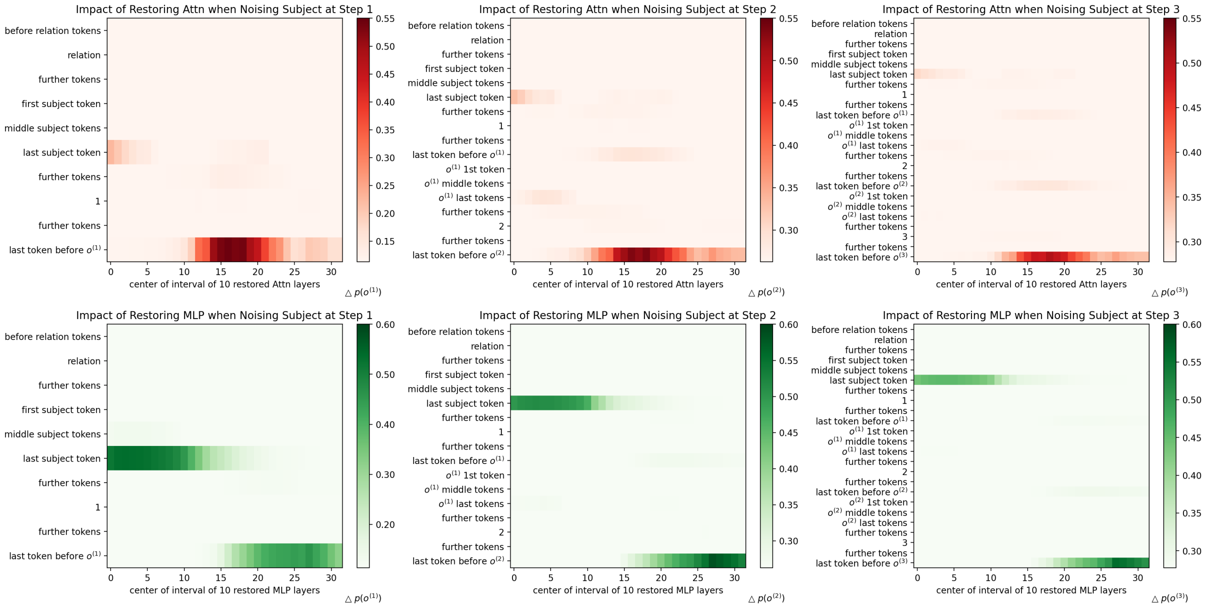

Figure 16: The impact of attention and MLPs' activations on LMs' predictions when intervening on the subject tokens across three answer steps macro-averaged across all models, templates, and 100 instances per dataset. Attention contributions dominate in the middle layers at the last token, while MLPs are important in early layers at the subject token and in late layers at the last token. The probability differences all peak around or above 0.55, reflecting the importance of the subject tokens.

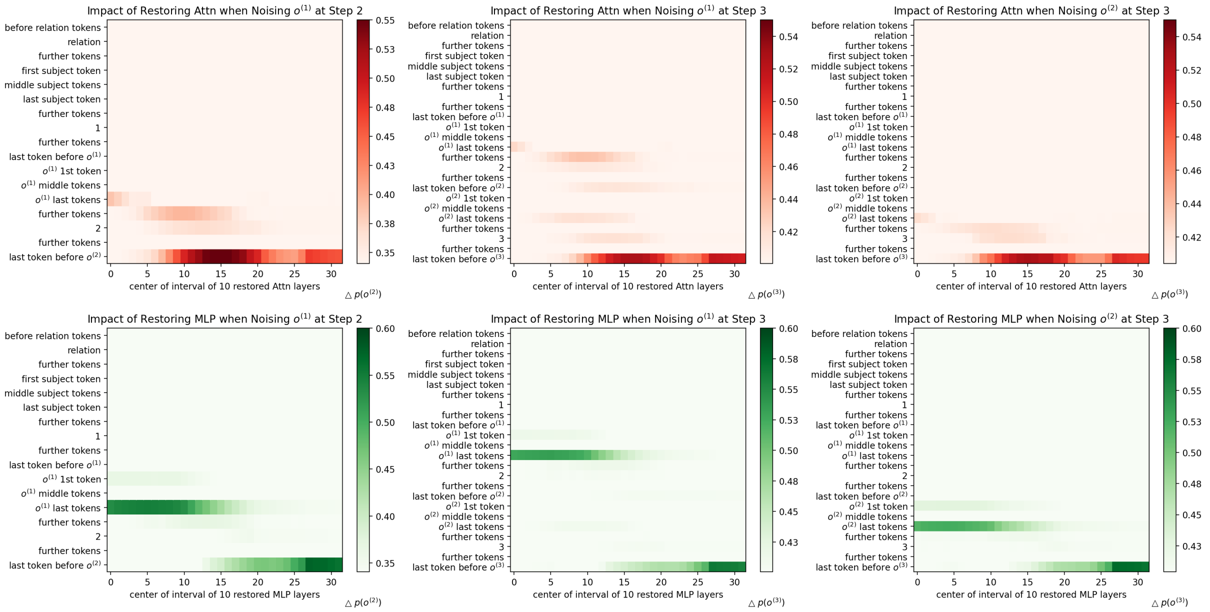

Figure 17: The impact of attention and MLPs' activations on LMs' predictions when intervening on previous answer tokens at step 2 and 3 macro-averaged across all models, templates, and 100 instances per dataset. Attention is important in both the middle and the last layers at the last token position. MLPs' contributions are critical in early layers at the previous answer positions and in final layers at the last token. The probability differences all peak around or above 0.54, indicating previous answer tokens are critical to models' predictions.

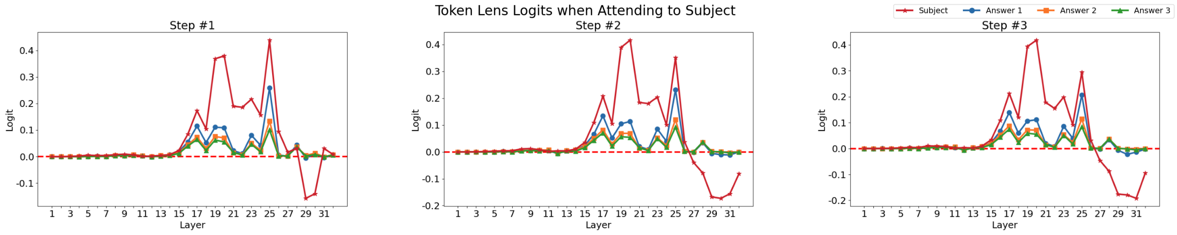

Figure 18: Token Lens logit values of subject and answer tokens across layers and answer steps when attending to the subject (macro-averaged across all datasets, models, and templates). Attention promotes and extracts subject information in the middle layers while suppressing it in later layers.

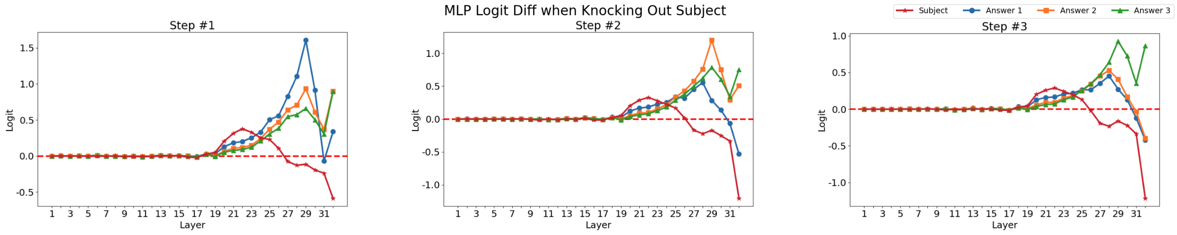

Figure 19: Logit differences of the subject and answer tokens between MLP outputs with and without knocking out attention from the last to the subject tokens (macro-averaged across all datasets, models, and templates). Positive logit differences for the answers and negative differences for the subject in later layers show that MLPs use the subject information to promote answers and suppress the subject.

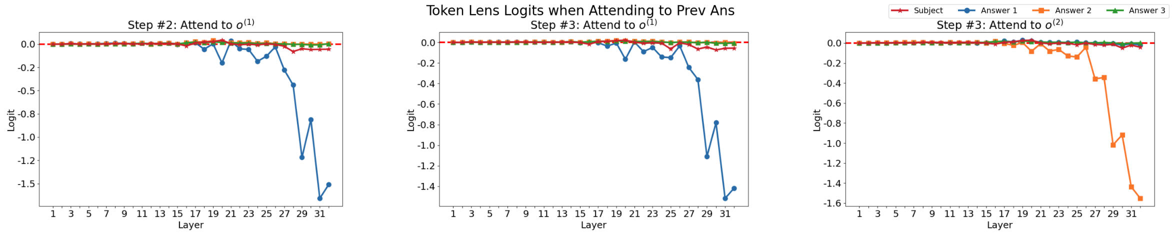

Figure 20: Token Lens logit values subject and answer tokens across layers and answer steps 2 and 3 (macro-averaged across all datasets, models, and templates) when attending to previous answers. The logit of the attended answer is negative at later layers, showing that the attention is suppressing previously generated answers.

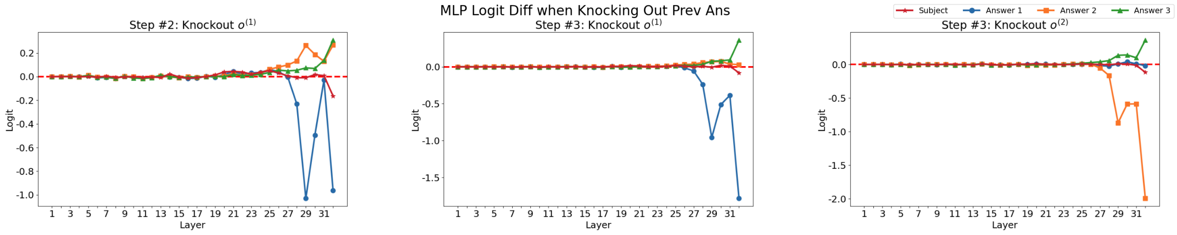

Figure 21: Logit differences for subject and answer tokens between MLP outputs with and without knocking attention from the last to previous answer tokens (macro-averaged across all datasets, models, and templates). All previously generated answer tokens have negative logits, and all new answers have positive logits. This result suggests that MLPs use previous answers for both repetition suppression and new answer promotion.

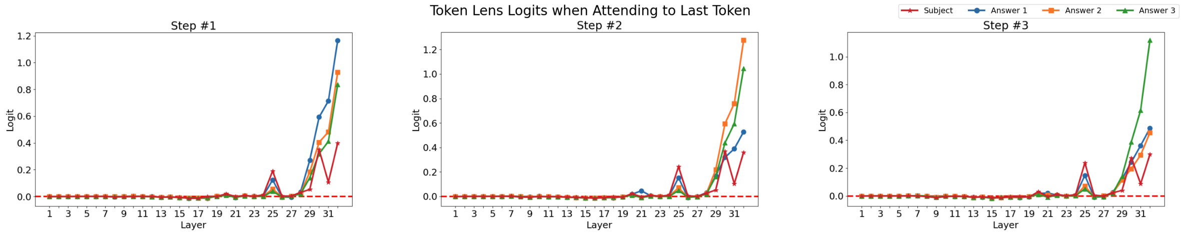

Figure 22: Token Lens logit values of subject and answer tokens across layers and answer steps when attending to the last token (macro-averaged across all datasets, models, and templates). Attention promotes all three answers and the subject at the final layers, with the answer for the current step having the highest logit.

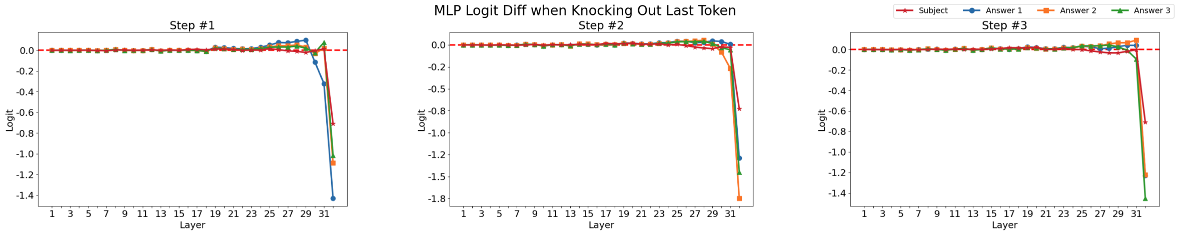

Figure 23: Logit differences for subject and answer tokens between MLP outputs with and without knocking attention from the last token to itself (macro-averaged across all datasets, models, and templates). The logit differences of all three answers and the subject are negative at the late layers, meaning MLPs output higher logits when it does not have information from the last token. This pattern may suggest a compensation behavior for the absence of direct attention to the last token to encourage the outputs to still be correct.

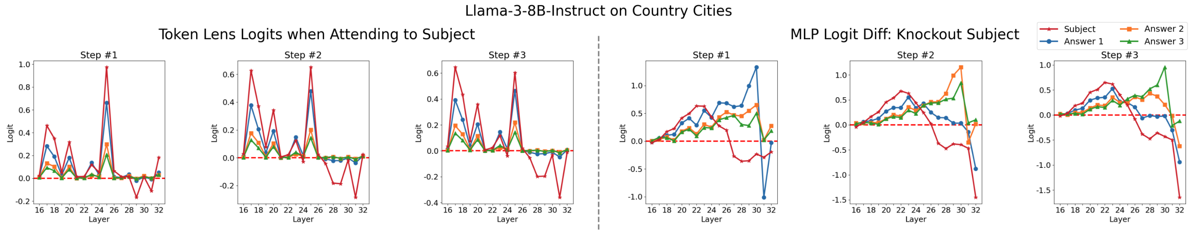

Figure 24: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens of Llama-3-8B-Instruct on Country-Cities dataset when attending to or knocking out the subject tokens (averaged across three prompt templates).

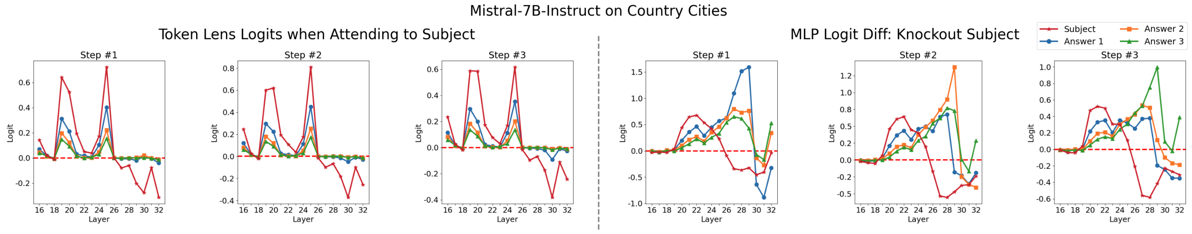

Figure 25: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens of Mistral-7B-Instruct on Country-Cities dataset when attending to or knocking out the subject tokens (averaged across three prompt templates).

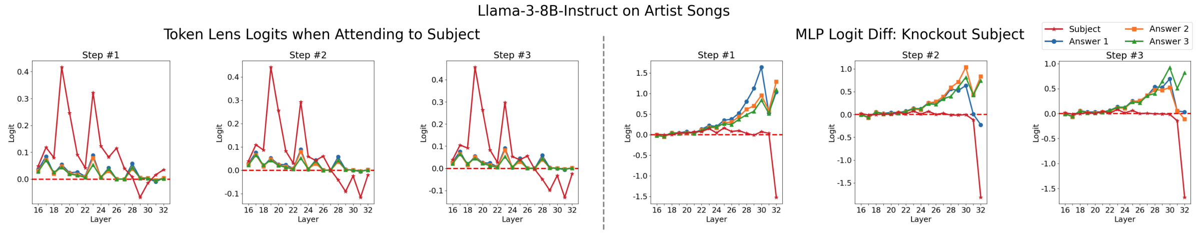

Figure 26: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens of Llama-3-8B-Instruct on Artist-Songs dataset when attending to or knocking out the subject tokens (averaged across three prompt templates).

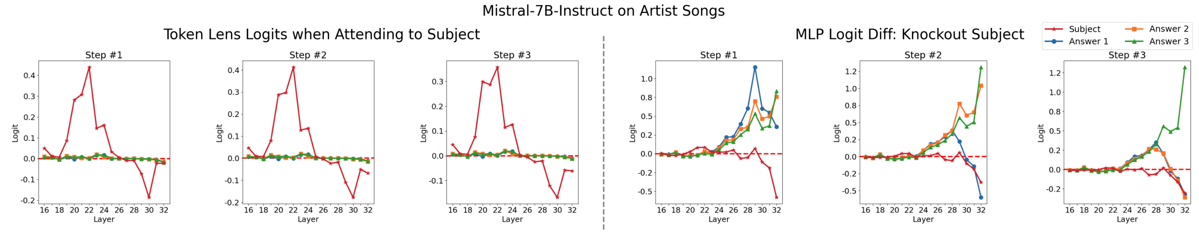

Figure 27: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens of Mistral-7B-Instruct on Artist-Songs dataset when attending to or knocking out the subject tokens (averaged across three prompt templates).

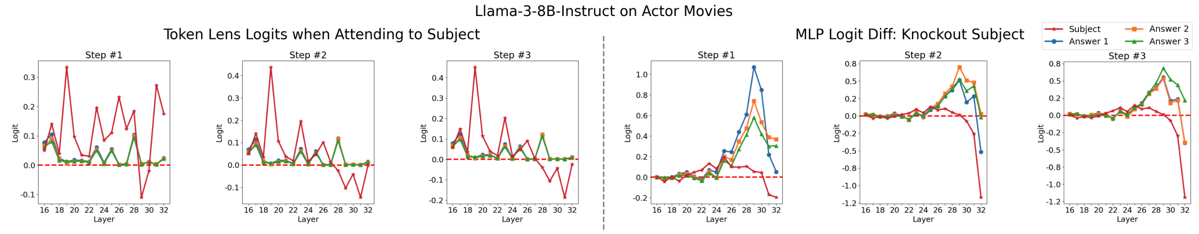

Figure 28: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens of Llama-3-8B-Instruct on Actor-Movies dataset when attending to or knocking out the subject tokens (averaged across three prompt templates).

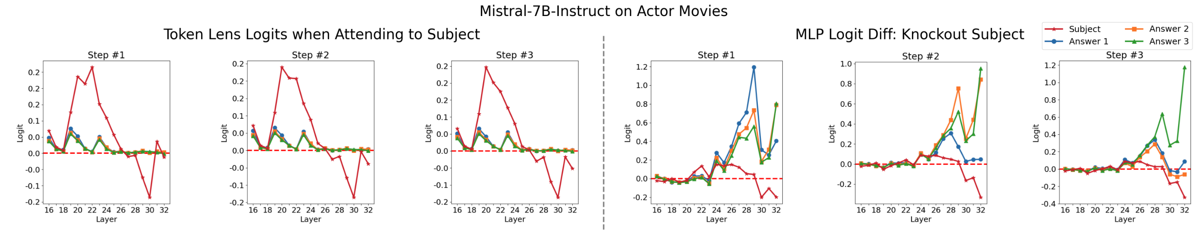

Figure 29: Token Lens logit values (left) and MLP logit differences (right) of subject and answer tokens of Mistral-7B-Instruct on Actor-Movies dataset when attending to or knocking out the subject tokens (averaged across three prompt templates).

Dataset	Model	Template 1	Template 2	Template 3
Country-Cities	Llama	List three cities from <country>	Name three cities located in <country>	Give the names of three cities in <country>
	Mistral	List the name of three cities from <country>	Provide just the names of three cities in <country>	State three city names from <country>
Artist-Songs	Llama	List three songs performed by <artist>	Name three songs sung by <artist>	Mention three tracks performed by <artist>
	Mistral	List three songs performed by <artist>	Provide just the names of three songs by <artist>	State three song titles by <artist>
Actor-Movies	Llama	List three movies acted by actor <actor>	Name three movies that feature <actor>	Mention three films that include <actor>
	Mistral	List the name of three movies acted by <actor>	Provide just the names of three movies featuring <actor>	State three movie titles starring <actor>

Table 2: Prompt templates used across datasets and models. Llama is short for Llama-3-8B-Instruct. Mistral is short for Mistral-7B-Instruct-v0.2.

Dataset	Model	Template 1	Template 2	Template 3
Country-Cities	Llama-3-8B-Instruct	122/168 (72.6%)	122/168 (72.62%)	123/168 (73.21%)
	Mistral-7B-Instruct-v0.2	116/168 (69.0%)	122/168 (72.62%)	116/168 (69.05%)
Artist-Songs	Llama-3-8B-Instruct	261/2077 (12.6%)	287/2077 (13.82%)	279/2077 (13.43%)
	Mistral-7B-Instruct-v0.2	206/2077 (9.9%)	240/2077 (11.56%)	217/2077 (10.45%)
Actor-Movies	Llama-3-8B-Instruct	1285/8790 (14.6%)	1263/8790 (14.37%)	1157/8790 (13.16%)
	Mistral-7B-Instruct-v0.2	965/8790 (11.0%)	905/8790 (10.30%)	528/8790 (6.01%)

Table 3: Number of correct cases and accuracy of two models on each dataset and template.

this threshold as model performance declined with more answer steps. See Tab. 4 for the accuracy of each model on every dataset.

Dataset	Llama-3-8B-Instruct	Mistral-7B-Instruct-v0.2
Country-Cities	92/167 (55.1%)	87/167 (52.1%)
Artist-Songs	82/2076 (4.0%)	74/2076 (3.6%)
Actor-Movies	582/7914 (7.4%)	422/7914 (5.3%)

Table 4: Number of correct cases and accuracy of two models on each dataset when the number of object entity $n = 5$.

We conducted token-level analyses with methods described in §6.1. We found that all major results and patterns at answer steps 4 and 5 match with those from answer steps 1, 2, and 3:

- Attention attends to subject tokens, promoting them in middle layers and suppressing them in deeper layers (Fig. 30, Fig. 31). MLPs use the subject to promote answers at the middle layers (Fig. 32, Fig. 33).

- Attention also attends to previous answers to suppress them (Fig. 34, Fig. 35), with MLPs reinforcing this suppression and promoting new answers in later layers (Fig. 36, Fig. 37).

- Attention at the last token promotes answers in the final layers (Fig. 38, Fig. 39), and MLPs compensate answer promotion when direct attention to the last token is intervened (Fig. 40, Fig. 41).

F.2 Ten Answer Steps

We also tried with 10 answer steps, but we could not collect 100 correct cases from any model and dataset. The model performance became much worse. For example, among all 154 Country-Cities data entries with at least 10 answers, Llama-3-8B-Instruct only got 56 correct, and Mistral-7B-Instruct-v0.2 only got 45 correct.

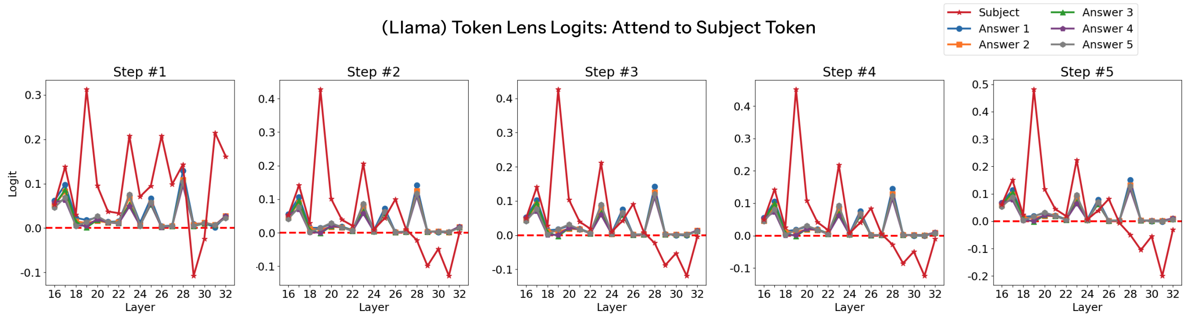

Figure 30: Llama's Token Lens logit values of subject and answer tokens across layers and answer steps when attending to the subject (macro-averaged across all datasets with template 1). Attention promotes and extracts subject information in the middle layers while suppressing it in later layers, which is consistent across all five answer steps.

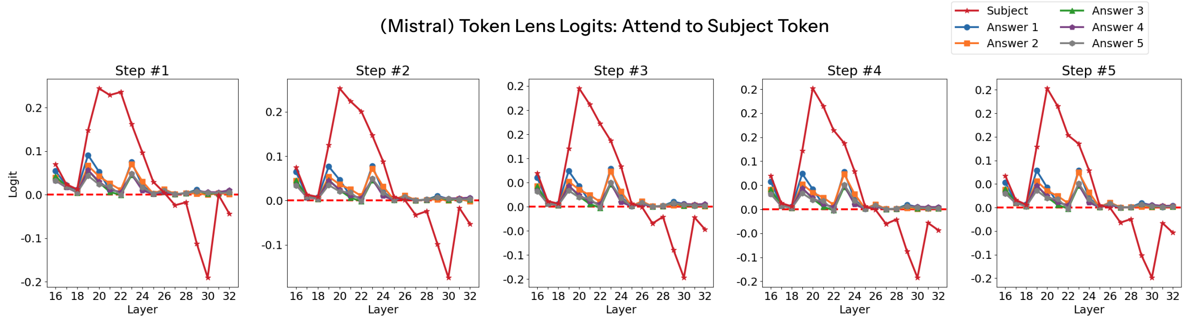

Figure 31: Mistral's Token Lens logit values of subject and answer tokens across layers and answer steps when attending to the subject (macro-averaged across all datasets with template 1). The patterns at steps 4 and 5 align with those observed in the first three steps.

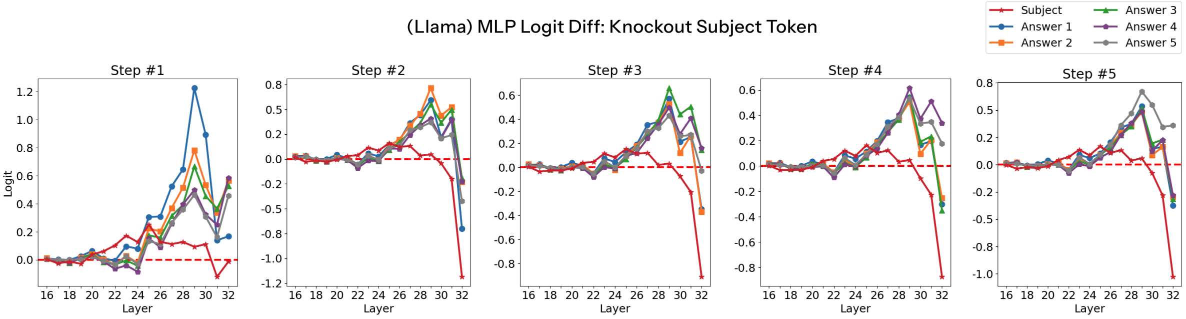

Figure 32: Llama's Logit differences of the subject and answer tokens between MLP outputs with and without knocking out attention from the last to the subject tokens (macro-averaged across all datasets with template 1). Positive logit differences for the answers and negative differences for the subject in later layers show that MLPs use the subject information to promote answers and suppress the subject. This pattern is consistent across all five answer steps.

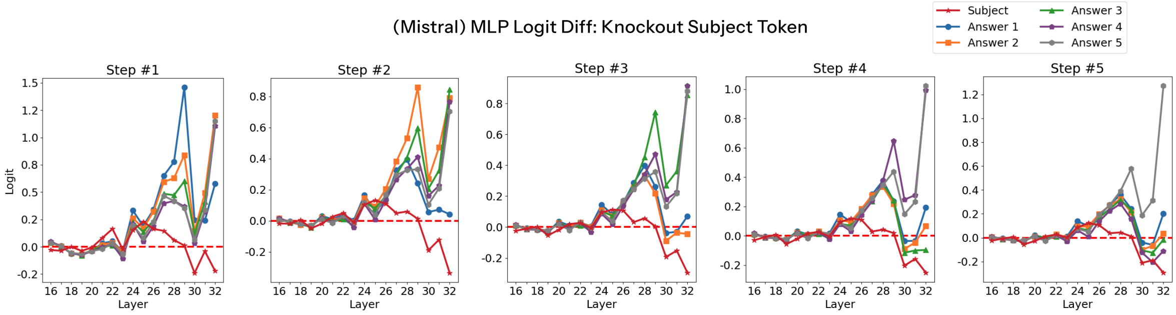

Figure 33: Mistral's logit differences of the subject and answer tokens between MLP outputs with and without knocking out attention from the last to the subject tokens (macro-averaged across all datasets with template 1). The patterns at steps 4 and 5 align with those observed from the first three steps.

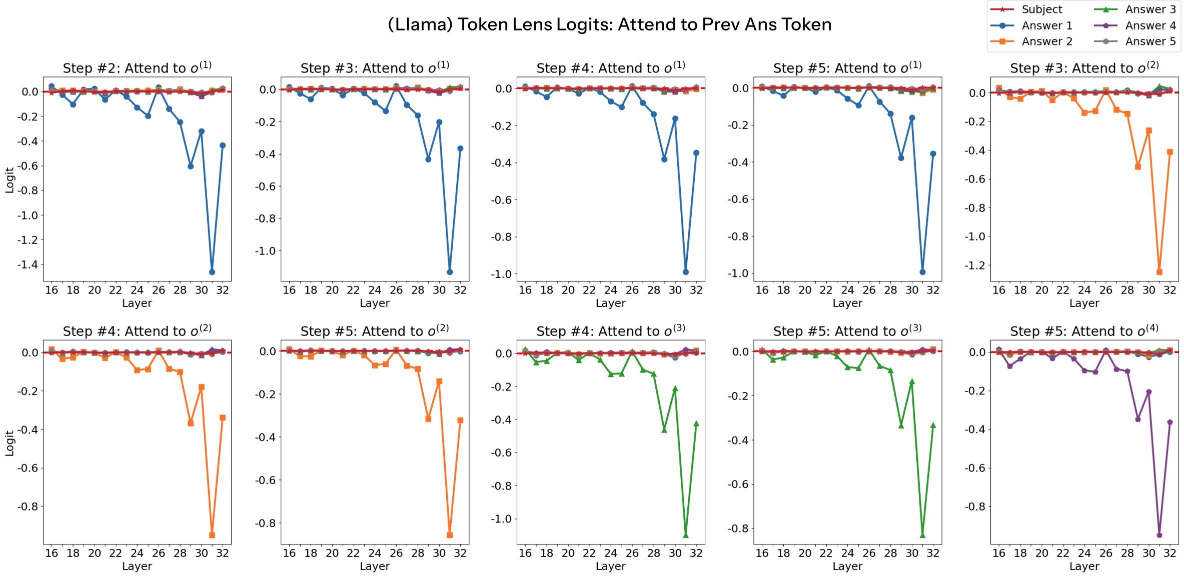

Figure 34: Llama's Token Lens logit values subject and answer tokens across layers and answer steps (macro-averaged across all datasets, models, and templates) when attending to previous answers. The logit of the attended answer is negative at later layers, showing that the attention is suppressing previously generated answers. The patterns at answer step 4 and 5 match with the ones discussed in the main section.

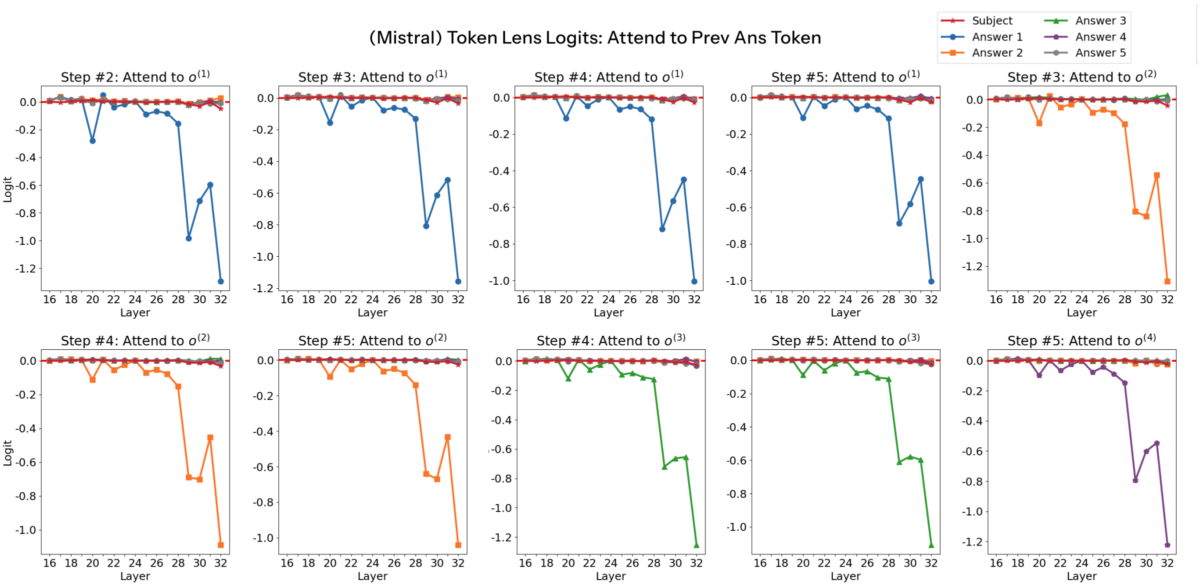

Figure 35: Mistral's Token Lens logit values subject and answer tokens across layers and answer steps (macro-averaged across all datasets, models, and templates) when attending to previous answers. The logit of the attended answer is negative at later layers, showing that the attention is suppressing previously generated answers. The patterns at steps 4 and 5 align with those observed in Llama and in the first three steps.

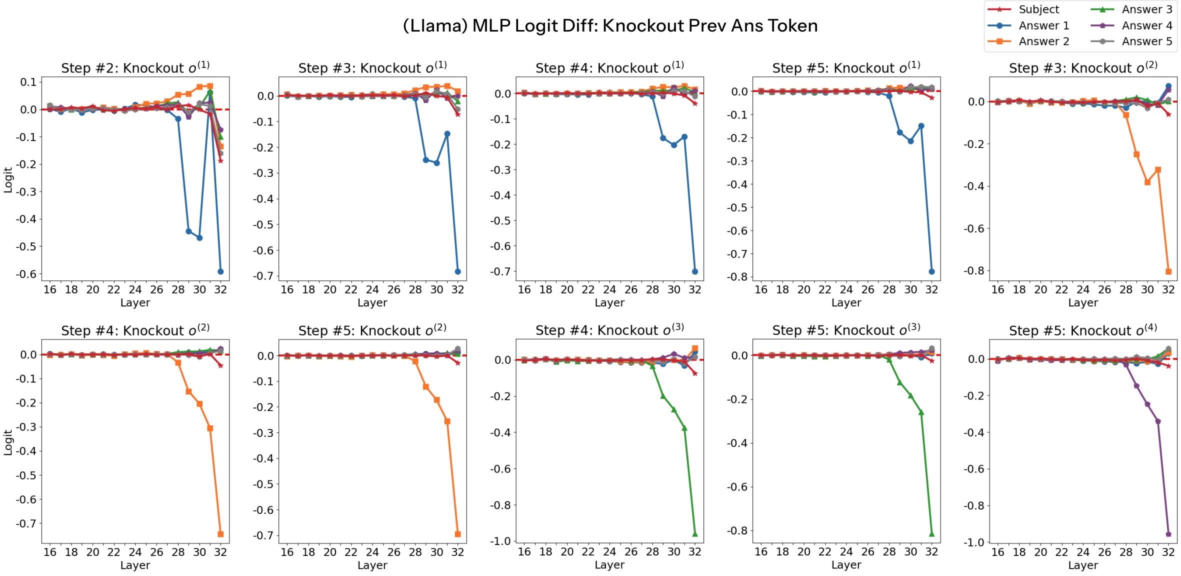

Figure 36: Llama's Logit differences for subject and answer tokens between MLP outputs with and without knocking attention from the last to previous answer tokens (macro-averaged across all datasets, models, and templates). All previously generated answer tokens have negative logits, and all new answers have positive logits. This result suggests that MLPs use previous answers for both repetition suppression and new answer promotion, aligning with the patterns discussed in the main section.

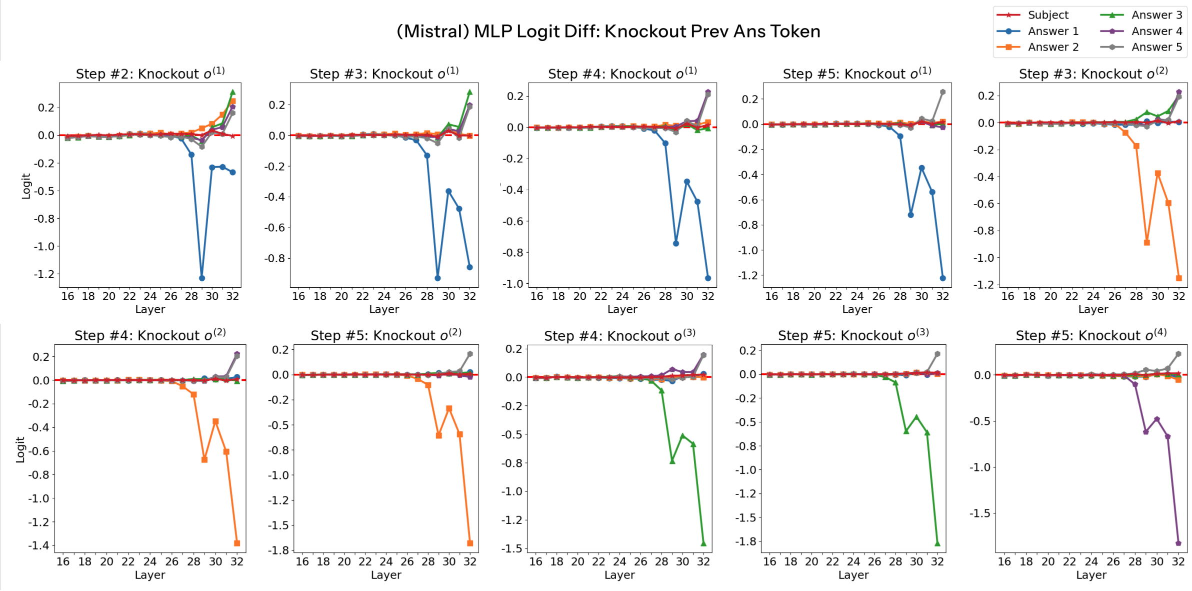

Figure 37: Mistral's logit differences for subject and answer tokens between MLP outputs with and without knocking attention from the last to previous answer tokens (macro-averaged across all datasets, models, and templates). The patterns at steps 4 and 5 align with those observed in the first three steps.

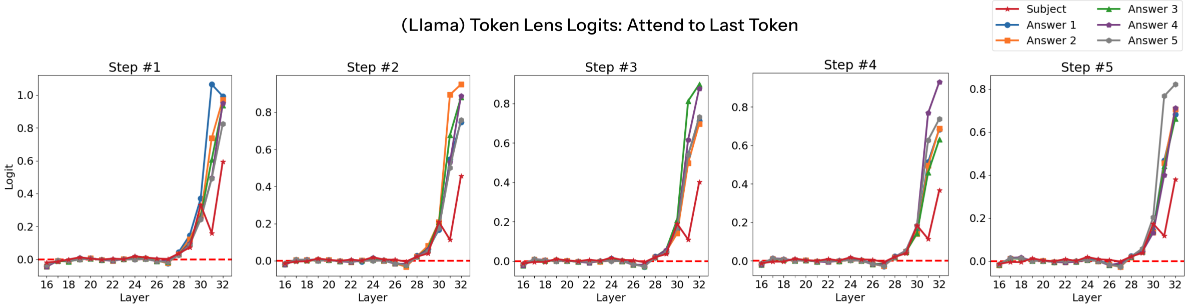

Figure 38: Llama's Token Lens logit values of subject and answer tokens across layers and answer steps when attending to the last token (macro-averaged across all datasets, models, and templates). Attention promotes all three answers and the subject at the final layers, with the answer for the current step having the highest logit, which align with the findings discussed in the main section.

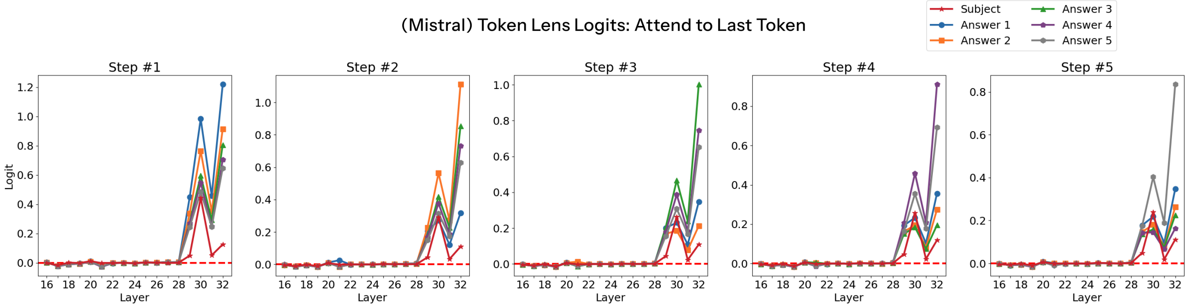

Figure 39: Mistral's Token Lens logit values of subject and answer tokens across layers and answer steps when attending to the last token (macro-averaged across all datasets, models, and templates). The patterns at steps 4 and 5 align with those observed in the first three steps.

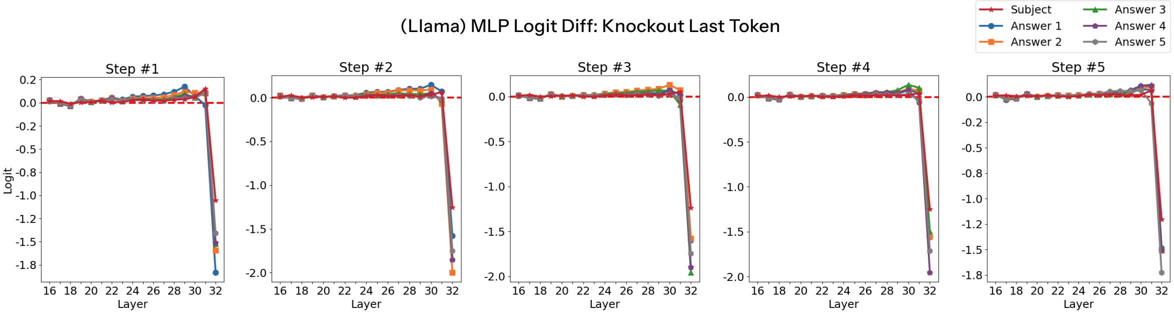

Figure 40: Llama's logit differences for subject and answer tokens between MLP outputs with and without knocking attention from the last token to itself (macro-averaged across all datasets, models, and templates). The pattern is aligned with those discussed in the main section.

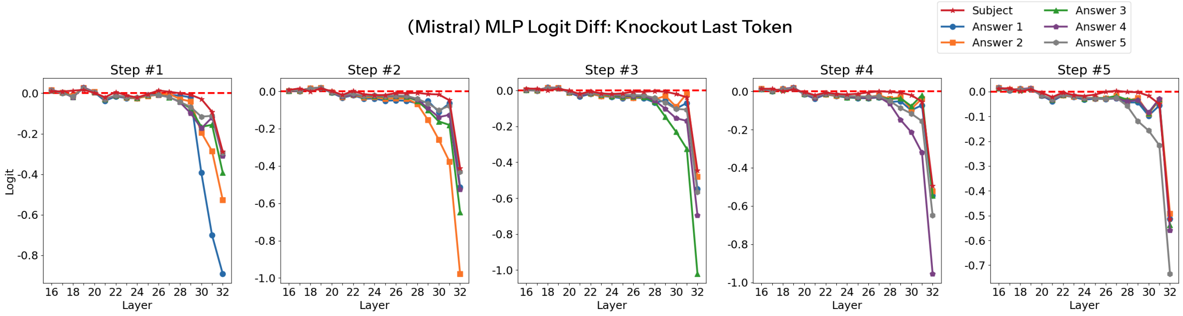

Figure 41: Mistral's logit differences for subject and answer tokens between MLP outputs with and without knocking attention from the last token to itself (macro-averaged across all datasets, models, and templates). The patterns at steps 4 and 5 align with those observed in the first three steps.