A Glitch in the Matrix?
Locating and Detecting Language Model Grounding with Fakepedia

Giovanni Monea,[◇] **Maxime Peyrard,**[♡] **Martin Josifoski,**[◇] **Vishrav Chaudhary,**[♠]
Jason Eisner,[♠] **Emre Kıcıman,**[♠] **Hamid Palangi,**[♠] **Barun Patra,**[♠] **Robert West**[◇]
[◇]EPFL [♡]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG [♠]Microsoft Corporation
{giovanni.monea, martin.josifoski, robert.west}@epfl.ch
{maxime.peyrard}@univ-grenoble-alpes.fr
{vchaudhary, jason.eisner, emrek, hpalangi, barun.patra}@microsoft.com

Abstract

Large language models (LLMs) have an impressive ability to draw on novel information supplied in their context. Yet the mechanisms underlying this contextual grounding remain unknown, especially in situations where contextual information contradicts factual knowledge stored in the parameters, which LLMs also excel at recalling. Favoring the contextual information is critical for retrieval-augmented generation methods, which enrich the context with up-to-date information, hoping that grounding can rectify outdated or noisy stored knowledge. We present a novel method to study grounding abilities using Fakepedia, a novel dataset of counterfactual texts constructed to clash with a model's internal parametric knowledge. We benchmark various LLMs with Fakepedia and conduct a causal mediation analysis of LLM components when answering Fakepedia queries, based on our Masked Grouped Causal Tracing (MGCT) method. Through this analysis, we identify distinct computational patterns between grounded and ungrounded responses. We finally demonstrate that distinguishing grounded from ungrounded responses is achievable through computational analysis alone. Our results, together with existing findings about factual recall mechanisms, provide a coherent narrative of how grounding and factual recall mechanisms interact within LLMs.

1 Introduction

One of the key factors underlying the massive success of large language models (LLMs) is their ability to encode and effectively recall a wealth of factual knowledge stored in their parameters (Heinzerling and Inui, 2021; AlKhamissi et al., 2022; Meng et al., 2023a). What elevates LLMs beyond promptable static repositories of knowledge is their capacity to adapt to new information and instructions provided in the context (Brown et al., 2020). Ideally, the LLM should integrate information from the context with static parametric knowledge to provide responses that robustly align with the intention specified in the prompt.

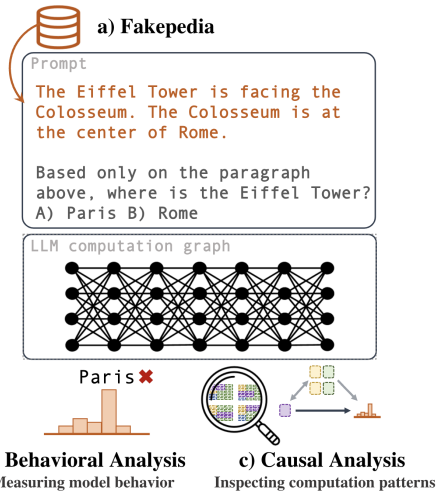

Figure 1: **Studying Grounding in LLMs.** This work makes three distinct contributions: (a) introducing a counterfactual dataset designed to measure the abilities of LLMs to ground their answer in the new information provided in the prompt, (b) conducting a descriptive analysis of grounding performances across several LLMs, and (c) implementing an improved causal mediation analysis that we use to show that computation patterns inside LLMs can predict whether the answer is grounded.

However, the in-context information can often contradict the internal parametric knowledge (Yu et al., 2023). Since the internal knowledge reflects only a partial snapshot of the state of the world at training time, enriching the context with up-to-date factual information is one of the most promising directions to keep the model relevant in a changing world (Hu et al., 2023; Yao et al., 2022). This is the main idea behind retrieval-augmented generation (RAG) (Lewis et al., 2020; Borgeaud et al., 2022;

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6828–6844

Mialon et al., 2023). However, this puts the model in a state of tension between internal factual recall and in-context grounding (from the retriever); thus, the success of such prompting methods hinges on the model's ability to accurately decide when to ignore its internal parametric knowledge. Yu et al. (2023) argued recently that LLMs, especially larger ones, often prefer their parametric knowledge.

Motivated by these observations, our work aims to study the mechanisms involved in grounding and factual recall for the practically important scenario where the two conflict. We hypothesize that different modes of information processing are triggered when the LLM engages in factual recall from its parameters compared to when it grounds generation in the context, and that these modes can be distinguished by the patterns of neural activation.

Previous works studying LLMs behavior typically craft experimental scenarios that isolate the behavior of interest. For instance, to test factual recall abilities, an LLM could be prompted to continue the sentence: `The Eiffel Tower is located in the city of ...`. Answering `Paris` suggests that the model *knows* and can recall the true fact. Yet, such behavioral analysis alone cannot reveal the underlying mechanisms that drive the observed behavior (Jain and Wallace, 2019; Teney et al., 2022; Bolukbasi et al., 2021). Deeper understanding requires opening the black box and examining the low-level computation patterns that give rise to the high-level behavior. While early studies have identified *computational correlates* between intermediate activations of a model and its outputs (Peyrard et al., 2021; Oba et al., 2021; Dai et al., 2022), there has been growing consensus that generalizable mechanistic explanations should stem from causal analysis rooted in interventionist experimental setups (Woodward, 2003; Potochnik, 2017; Pearl and Mackenzie, 2018; Geiger et al., 2022a, 2023). Building on these insights, several studies have developed rigorous methods to intervene on model components, putting the model in counterfactual states and systematically measuring the impact of these interventions on the model's behavior (Geiger et al., 2022b; Wu et al., 2023b; Meng et al., 2023a; Geva et al., 2023). These experimental setups have allowed researchers to form a clearer picture of the factual recall mechanisms (Geva et al., 2021; Wallat et al., 2020; Meng et al., 2023a; Kobayashi et al., 2023; Geva et al., 2023).

Grounding complements factual recall, but has received much less scrutiny. In this work, we propose a thorough analysis of the grounding mode of computation of several LLMs, and make four important contributions:

- A new counterfactual dataset, called **Fakepedia**, crafted especially to isolate grounding behavior from factual recall by setting LLMs in tension between the two modes (Sec. 4).
- A descriptive behavioral analysis of several LLMs measuring their grounding abilities in the challenging task presented by Fakepedia (Sec. 5).
- A new causal mediation analysis, called Masked Grouped Causal Tracing (**MGCT**), that assesses the effect of subsets of the model states on the model's behavior (Sec. 6.1).
- A set of findings coming from applying MGCT to Llama2-7B and GPT2-XL on Fakepedia (Sec. 6.2). Specifically, we show that (i) contrary to factual recall, grounding is a distributed process, (ii) the activity of a few MLPs differs significantly between grounded vs. ungrounded modes, and (iii) we can predict whether the model is engaged in grounding behavior by looking only at the computation graph, with an accuracy of 92.8% (Sec. 7).

To support further research in the space of grounding and in-context learning, we release the Fakepedia dataset and the code pipeline to reproduce or extend it. We also release a user-friendly implementation of MGCT that can be employed to study the computation patterns of LLMs when engaged in any other behavior of interest.[1]

2 Related Work

Understanding the inner workings of LLMs poses a significant challenge, given their complex architectures (Carvalho et al., 2019; Rogers et al., 2020; Geiger et al., 2023). To better analyze models, controlled datasets are often crafted that isolate the target behavior. Behavioral experiments are then supplemented by a deeper inspection of the underlying low-level computation patterns that explain the observed behavior. This section gives a brief overview of previous papers relevant to this work.

[1]Code and dataset are available at https://epfl-dlab.github.io/llm-grounding-analysis

2.1 Mechanistic Interpretability

Discovering the low-level mechanisms that give rise to high-level behaviors has become an important goal for research in AI interpretability and allowed for better understanding and prediction of models behavior (Mu and Andreas, 2020; Geiger et al., 2022b; Wu et al., 2023b; Vig et al., 2020; Dai et al., 2022; Oba et al., 2021; Peyrard et al., 2021). Understanding these mechanisms allows us to better predict out-of-distribution behavior (Mu and Andreas, 2020; Geiger et al., 2022b; Wu et al., 2023b), identify and fix errors (Vig et al., 2020). For instance, Dai et al. (2022) analyzed BERT activations revealing that some neurons are positively correlated with specific facts. Similarly, Oba et al. (2021) demonstrated associations between neuron activations and specific phrases in model output. Additionally, in a matched study on humor detection, Peyrard et al. (2021) identified attention heads in BERT encoding the funniness of sentences.

To go beyond statistical correlations, a promising approach is to view the Transformer's computation graph as a causal graph (Elhage et al., 2021; Meng et al., 2023a; Geiger et al., 2023; Wu et al., 2023a). Targeted interventions on the computation are then applied to estimate the impact of individual components on model behavior (Modarressi et al., 2022; Mohebbi et al., 2023; Wang et al., 2022; Nanda et al., 2023; Merullo et al., 2023; Belrose et al., 2023; Meng et al., 2023a; Vig et al., 2020). In particular, causal mediation analysis applied to components of GPT-2 has revealed that some multilayer perceptrons (MLPs) are key-value stores of factual knowledge (Geva et al., 2021; Wallat et al., 2020; Meng et al., 2023a; Kobayashi et al., 2023). Several works have been able to even edit factual knowledge directly in the weights of pre-trained Transformers (Meng et al., 2023b; Mitchell et al., 2022; De Cao et al., 2021). With a similar interventional setup, Geva et al. (2023) studied information flow during factual knowledge recall, finding critical aggregation points, especially located in a few attention heads. Interestingly, Haviv et al. (2023) demonstrated the critical role of MLPs in early layers when the model is recalling memorized sequences.

While recalling factual knowledge relates to memorization, in this work we study the ability of models to ground their answers based on information in the context, i.e., to incorporate new information not seen at training time. In particular, we craft scenarios that set the LLMs in tension between the mode of factual recall and the mode of grounding. This contributes to the broader discussion on generalization versus memorization (Razeghi et al., 2022; Kandpal et al., 2023; Hupkes et al., 2023; Haviv et al., 2023; Xu et al., 2024). However, grounding remains much less studied than factual recall. In a contemporary study, Yu et al. (2023) also inspect the problem of grounding using mechanistic interpretability methods. Their analysis focuses on the role of attention heads when forcing the model to ground its answer in the prompt. Our findings nicely complement theirs. When combined with existing findings about factual knowledge and information flow during recall, our results begin to portray a coherent narrative that we detail in Sec. 8.

2.2 Counterfactual Datasets

The necessity for counterfactual datasets is becoming increasingly evident in contemporary research.

However, producing counterfactual datasets is not straightforward. Works like those by Neeman et al. (2022) and Zhou et al. (2023) adopt the methodology proposed by Longpre et al. (2022), which involves substituting entities within existing paragraphs. Li et al. (2022) proposes a similar method also based on entity substitution. Another approach involves generating new documents altogether: Köksal et al. (2023) and CH-Wang et al. (2023), for instance, build on the idea that a model tasked with producing a response may inadvertently generate counterfactual text.

Concurrently, language models have shown remarkable proficiency in synthetic dataset generation (Gunasekar et al., 2023; Schick and Schütze, 2021; Hartvigsen et al., 2022; Josifoski et al., 2023; Eldan and Li, 2023; Lingo, 2023). Building on these advancements, our research demonstrates that it is feasible to employ language models for generating novel, original, and high-quality counterfactual paragraphs from scratch.

3 Background

3.1 Terminology

We view *facts* as triplets made of a subject, a relation, and an object. A fact is said to be *true* if it aligns with our observed world. For example, `{Eiffel Tower | is located in | Paris}` is a *true* fact. In our experiments, we focus on true facts that the models can recall

and therefore *know*. By contrast, a *counterfactual* triplet is any triplet that does not form a true fact.

In our experiments, we test models by asking them to produce the object of a triplet, either directly as its next token or via a multiple-choice questionnaire (MCQ). This facilitates a systematic inspection of the model's answers by comparing them with the target object. It follows previous related work (Yu et al., 2023; Meng et al., 2023a).

3.2 Grounded vs. Factual

It is crucial to differentiate between a *factual* answer and a *grounded* answer. A factual answer is the object of a true fact triplet, while a grounded answer is the object triplet logically consistent with the information in the context of the prompt. Factuality pertains to the model's encoded knowledge and its ability to retrieve it, whereas *grounding* involves the model's capacity to adapt to its context and reason about new information.

While factual recall has been extensively studied in previous work, this work aims to study grounding. However, grounding and factual recall can be challenging to disentangle. For instance, when given a factual description about `Paris` in the prompt and asked for the location of the `Eiffel Tower`, the correct answer could arise from factual recall, grounding, or a mixture of both processes.

To isolate grounding from factual recall and related processes, we generate new counterfactual datasets. In these datasets, the grounded answer is always non-factual, implying that the context implicitly describes a false triplet. The context in Fig. 1 is one such example, implicitly placing the `Eiffel Tower` in `Rome`. If the model produces the grounded answer (i.e., `Rome`), it indicates that grounding processes occurred and made the model integrate information from and reason about the context. Conversely, if the model produces any other object (e.g., `Paris`, `Napoli`), it has not successfully integrated information from the context.

We also observe that it is possible to interpret cases where the models exhibit ungrounded behavior as (intrinsic) hallucinations. We further elaborate on this in Appendix A.1.

4 Counterfactual Data Creation

Our study requires datasets of counterfactual paragraphs to create prompting scenarios where the language model's parametric knowledge conflicts with the information in the context. We now describe the process of creating such datasets.

4.1 Counterfactual ParaRel

The datasets used in this study were derived from the ParaRel dataset (Elazar et al., 2021). The templates were modified to make them suitable for prompting LLMs, ensuring the generation of the object as the next token. The selection process retained triplets where GPT2-XL yielded the highest probability for the true object as the next token.

Counterfactual triplets were constructed by choosing alternative objects for each triplet, creating 21,308 counterfactual triplets in total. This dataset aims to challenge LLMs by emphasizing the tension between parametric knowledge and contextual grounding. For more details, readers are referred to Appendix B.1.

4.2 Fakepedia

Based on the counterfactual ParaRel dataset, we create Fakepedia, a collection of counterfactual paragraphs (as in Fig. 1), coming in two variants: **Fakepedia-base** and **Fakepedia-MH**, where MH stands for "Multi-Hop." The base variant contains, for each triplet in the counterfactual ParaRel, an LLM-generated paragraph that "entails" that triplet in the sense of Dagan et al. (2013). In the MH variant, the paragraph does not suggest the triplet as directly as in the base variant, but still logically implies it by a two-hop reasoning process, by way of an intermediary triplet that is also counterfactual.

We refer the reader to Appendix B.2 for two practical examples of Fakepedia paragraphs and more details on the dataset construction.

5 Descriptive Behavioral Analysis

We first inspect the behavior of several LLMs in the grounding challenge proposed by Fakepedia datasets.

Experimental setup. In this study, we use two prompt templates to query the model about the object of the triplet described by the Fakepedia paragraph. The first prompt template queries the model about the object of a triplet described in a Fakepedia instance using a multiple-choice question (MCQ) with two possible answers: (i) the grounded answer being the target (counterfactual) object described or implied in the Fakepedia text, and (ii) the factual answer being the object in the true triplet. The prompt explicitly instructs the

Table 1: **Grounding accuracy on Fakepedia for various LLMs.** The 'Instruction' column refers to whether the prompt explicitly instructed the models to rely only on the context to answer.

		Mistral	Zephyr	Llama2			GPT-3.5 Turbo			GPT-4 Turbo
Dataset	Instruction	7B	7B	7B	13B	70B	03/01	06/13	11/06	11/06
FP	With	**92%**	58%	22%	84%	90%	61%	54%	50%	28%
	Without	90%	52%	1%	70%	73%	47%	24%	27%	1%
FP-MH	With	60%	10%	4%	**82%**	71%	7%	8%	10%	50%
	Without	49%	8%	0%	58%	50%	3%	2%	2%	5%

model to base its answer solely on the context. To mitigate potential ordering bias, we create two versions of each MCQ, reversing the order of options.

The second prompt template does not explicitly instruct the model to use only the context to answer the question.

The results, presented in Table 1, report the percentage of instances in which the models correctly select the grounded answer. The random baseline has an accuracy of 50%.

Analysis. Predictably, the prompting scheme explicitly instructing models to rely solely on context makes models more often choose the grounded answer. Also, Fakepedia-MH, which necessitates reasoning about information within the context, poses a greater challenge for models overall, except for GPT-4 Turbo. In this case, although the 50% grounding accuracy may be an indication that the model is randomly guessing at each step, it also shows that GPT-4 Turbo tends to be more compliant and less critic when logic reasoning is required.

Notably, we observe surprisingly poor grounding from GPT-4 Turbo, with a rate of only 1% and 5% with the less explicit prompt. The accuracies are worse than random guessing (50%), suggesting a clear preference for the model's parametric knowledge. A trend in this direction is also apparent across successive snapshots of GPT-3.5 Turbo.

In the Llama2 series, the 7B model also exhibits a strong preference for its internal parametric knowledge, but starting from 13B, the models distinctly favor the grounded answer.

Mistral-7B emerges as the most compliant model, robustly selecting the grounded answer in Fakepedia. Notably, Mistral-7B is the most grounded model even when it is not specifically instructed to answer based on the context. In Fakepedia-MH, while the performance drops, Mistral-7B remains above chance-level when instructed to remain grounded.

Overall, complex patterns emerge as models exhibit diverse behaviors in different scenarios. Contrary to Yu et al. (2023), who found that larger models tend to favor their parametric knowledge more than smaller models, our findings introduce nuances. For instance, in the Llama2 series, larger models prove significantly more accurate than the 7B model, with little difference between 13B and 70B.

6 Causal Analysis of the Computation Graph

In this section, our objective is to investigate whether measurable patterns within the computation graph of LLMs can effectively differentiate between grounded and ungrounded behaviors.

In the context of LLM interpretability, addressing this question involves intervening on model representations to create counterfactual model states, and then systematically studying the effects of these interventions on model behavior (Ghandeharioun et al., 2024; Geiger et al., 2022a; Vig et al., 2020; Meng et al., 2023a; Goyal et al., 2019; Feder et al., 2021).

In this work, we generalize one such method, causal tracing (Meng et al., 2023a), to improve its robustness and efficiency—resulting in what we call *Masked Grouped Causal Tracing* (MGCT), whose execution is depicted in Fig. 2. We then apply MGCT to relate low-level computation patterns of LLaMA-7B and GPT2-XL against their observed grounding behavior when answering queries from Fakepedia.

6.1 MGCT Analysis

Causal tracing. The execution of a transformer forward pass yields a causal graph describing the

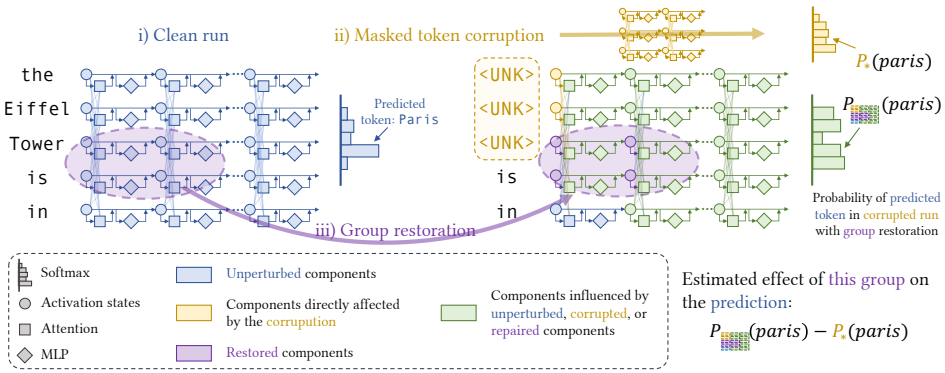

Figure 2: **Masked Grouped Causal Tracing (MGCT).** This figure illustrates the mediation analysis from MGCT, refining the preceding causal tracing method from Meng et al. (2023a). The process involves three steps: **(i) Clean run:** all states within the computation graph are recorded during a forward loop, resulting in a predicted token, in this case "Paris". **(ii) Corrupted run:** the subject tokens are substituted with special non-textual tokens such as <UNK> or <EOS>, leading to a distinct probability for the predicted token. **(iii) Restored run:** the corrupted with the restoration of a group of states (in this instance, four hidden activations) to their values from the clean run, resulting in a partially restored probability for the predicted token. The indirect effect is estimated by the extent to which the restoration of these states contributes to the probability restoration of the predicted token.

dependencies among states in the computation until reaching the softmax output probabilities. Specifically, given a sequence of k input tokens, it yields a grid of hidden states $h_k^{(l)}$ which is obtained from the previous layer $(l-1)$ by adding previous hidden states $h_k^{(l-1)}$ (residual connections), a global attention $a_k^{(l)}$ (attention head), and a local MLP contribution $m_k^{(l)}$, according to the following process (Vaswani et al., 2017):

$$h_k^{(l)} = h_k^{(l-1)} + a_k^{(l)} + m_k^{(l)} \quad (1)$$

$$a_k^{(l)} = \text{attn}^{(l)}\left(h_0^{(l-1)}, h_1^{(l-1)}, \ldots, h_k^{(l-1)}\right) \quad (2)$$

$$m_k^{(l)} = \text{FF}\left(a_k^{(l)} + h_k^{(l-1)}\right) \quad (3)$$

where FF is a two-layer feed-forward MLP.

The causal tracing method proposed by Meng et al. (2023a) is a mediation analysis that estimates the **average indirect effect** of individual components on predictions made by the LLMs. The method involves the following steps:

(i) A **clean run** that records every state during a forward call. The model's prediction is the most probable next token o with probability $P(o)$.

(ii) A **corrupted run** follows, where some prompt tokens are perturbed by adding noise to their embeddings. The forward computation is then executed with the corrupted input, often resulting in a different output state and a distinct probability $P_*(o)$ for the output token o.

(iii) A **partially-repaired** run ensues, where the corrupted run is re-executed, except for a selected state $h_k^{(l)}$ at a specific layer l and token k. The state $h_k^{(l)}$ is restored to the value it had in the clean run, and the computation continues. This produces a new probability for the output token o, denoted as $P_*^{clean(h_k^{(l)})}(o)$.

The estimated indirect effect of the mediating state $h_k^{(l)}$ is then given by $P_*^{clean(h_k^{(l)})}(o) - P_*(o)$. Aggregating over different inputs provides the estimated average indirect effect of the mediating state $h_k^{(l)}$ on the model's predictions. The mediation analysis is repeated for every state in the LLM architecture to obtain a global map of indirect effects.

Meng et al. (2023a) focus on the effects of restoring attention $a_k^{(l)}$, MLP $m_k^{(l)}$, and hidden activation states $h_k^{(l)}$ separately within each Transformer block.

Masked causal tracing. In our experiments, we observed that the outcome of the causal tracing algorithm is highly sensitive to the choice and magnitude of noise in corrupted runs. To enhance the robustness and generalizability of causal tracing, we propose to perturb directly the tokens instead of the embeddings. This corruption method involves substituting tokens with special non-textual tokens, such as the UNK or EOS tokens. This modification results in more stable masked causal traces, avoids the need for multiple runs to average out the ran-

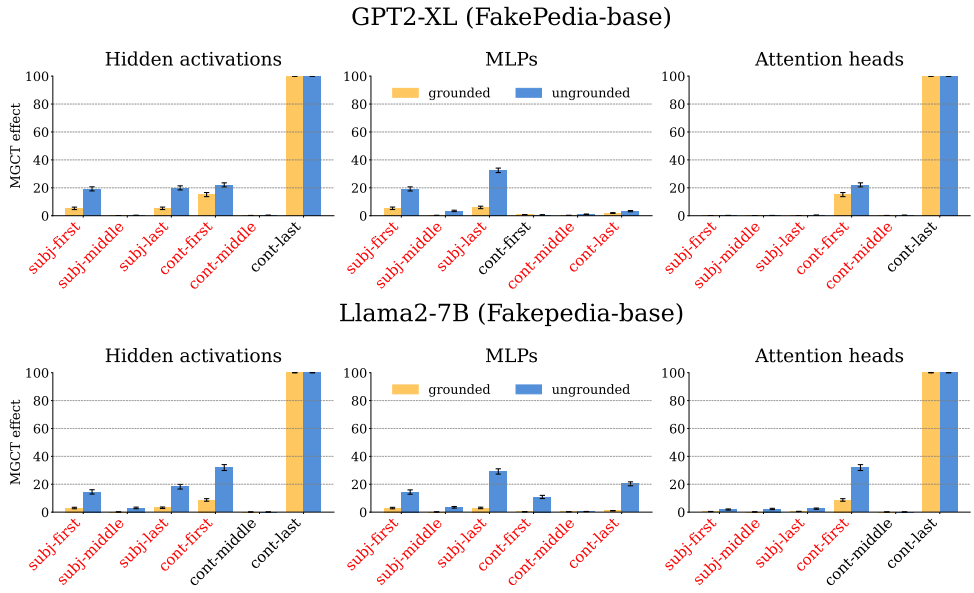

Figure 3: **Masked Grouped Causal Tracing analysis on Fakepedia-base.** This figure illustrates the application of MGCT analysis to GPT2-XL and Llama2-7B on the Fakepedia-base dataset. We distinguish between instances where the models generated grounded answers and those where they generated ungrounded answers. In the MGCT analysis, we restore full columns together, all states across all layers for a given column at a time, resulting in one effect per token. On the y-axis, we report the percentage of explained change in probability between the clean and corrupted runs due to the restoration of the column. To average across different sequences, we bucketed tokens into subject (subj-) categories and following tokens in the prompt (cont-). Red labels on the x-axis indicate that the difference in MGCT effect between grounded and ungrounded responses is statistically significant based on a t-test with a p-value threshold of 0.01.

domness, and consequently requires less memory and increases the speed of the process.

Grouped causal tracing. The causal tracing method repairs one state at a time, necessitating $L \times K$ mediation analyses, one per attention block, where L is the number of layers and K is the number of tokens to be restored.

Additionally, we might be interested in the measuring the joint effect of a group of states. Unfortunately, group effects cannot be estimated simply by aggregating the individual effects of states in the group, due to complex non-linear interactions among the states.

To extend the method's generality, we propose repairing groups of states simultaneously. Formally, the group is defined by a binary filter of dimensionality $L \times K$, one entry per attention block to be restored. The binary indicator at position (l,k) in the filter determines whether the state in the (l,k) block should be restored in the ongoing mediation analysis. Different mediation analyses can be executed by using different filters.

This approach allows flexible control. For instance, one can use patches of $M \times N$ states, applied at different locations in the architecture, akin to convolutional filters, allowing for overlap or not between groups by adjusting the stride and patch size. Fig. 2 illustrate MGCT with a patch of size 2×2, i.e., 4 states being repaired at once.

Futhermore, using groups with more than one state increases efficiency when aiming to cover all states. In MGCT, the number of mediation analyses is the number of filters. For instance, with non-overlapping patches of size $M \times M$, the number of mediation analyses to cover all states is reduced by a factor of M^2 compared to single-state restorations. In our experiments involving grounding (described below), we achieved a 48× speed-up for GPT2-XL and a 32× speed-up for LLaMA-7B, without compromising our ability to predict the high-level behavior of grounded vs. ungrounded (in Sec. 7).

6.2 MGCT Experiments

Experimental setup. We perform MGCT analysis on GPT2-XL (Radford et al., 2019) and Llama2-7B (Touvron et al., 2023) using the Fakepedia dataset. Data points were categorized as grounded or ungrounded based on model answers, considering only the first token for instances with multiple tokens. The analysis involved simultaneous restora-

tion of all states across all layers for a specific token, reducing the computational time by orders of magnitude compared to Meng et al. (2023a). This approach effectively examines the impact of each token on the output.

The MGCT effect quantifies the percentage of explained probability between clean and corrupted runs. Results for Fakepedia-base are reported in Fig. 3. Further details, including Fakepedia-MH results, can be found in Appendix D.1.

High effect of MLPs on ungrounded answers. Our MGCT analysis reveals clear differences in the effect patterns between cases where the model is grounded compared to when it is not grounded. There are many types of tokens and types of states for which the MGCT effect difference between grounded and ungrounded is statistically significant. This indicates the existence of distinct computation processes when the model derives its response by grounding it in the prompt text versus relying solely on its internal memory. Notably, it is clear that the MLPs' activations, particularly on the last subject token, have a high effect when producing ungrounded answers. Our results nicely combine with the findings from Meng et al. (2023a), who demonstrated that Transformers' MLPs serve as repositories of factual knowledge. In our context, this suggests that when MLPs heavily influence model responses, they are retrieving factual knowledge from memory rather than reasoning about the current information in the prompt.

Furthermore, Geva et al. (2023) recently analyzed information flow when the model is recalling factual knowledge from its memory and found the last subject token to be a crucial step in the information aggregation pipeline. In our context of grounding, we also find strong evidence that critical information processing is happening in this token position.

While Yu et al. (2023) found that intervening on a few attention heads can switch the model from an ungrounded to a grounded behavior, it appears that when engaged in grounding no single component emerges as having a strong impact on the prediction. This seems to indicate that, contrary to factual recall, grouding may be a more distributed process without a clear localization.

Interestingly, we find that GPT2 and Llama2 exhibit similar patterns, showing similar levels of causal effect for the same buckets of tokens. In our early experiments, we used LLaMA (Touvron et al., 2023) instead of Llama2 and we found that LLaMA presented a more distributed pattern (see Fig. 5 in Appendix D.1). Surprisingly, Llama2 behavior differs from it and resembles GPT2.

7 Automatic Detection of Ungrounded Responses

Within our MGCT analysis, we identify distinct computation patterns between grounded and ungrounded responses. To automate detection, we create a balanced dataset with 4,000 GPT2-XL responses from the Fakepedia-base dataset. Employing an 80%–20% train–test split, we extract 18 features from MGCT outputs for a binary classifier (e.g., attention, MLP, hidden activation effects, by grouping tokens results as in the MGCT mediation analysis). After training an XGBoost classifier, we achieve a 92.8% accuracy on the test set. Feature importance analysis identifies MLPs on the last subject tokens as most crucial, aligning with our MGCT findings. Our study demonstrates that distinguishing grounded from ungrounded responses is achievable through computation analysis alone. We refer to Appendix E for a more detailed description of the experimental setup.

8 Discussion

Extensive research has studied factual recall, producing several significant insights. Specifically, LLMs exhibit the capacity to store and retrieve factual knowledge. This knowledge is localized within a few MLPs functioning as distributed key-value databases (Geva et al., 2021; Wallat et al., 2020; Meng et al., 2023a; Kobayashi et al., 2023; Meng et al., 2023b; Mitchell et al., 2022; De Cao et al., 2021). Few attention heads are known to be crucial for information routing during factual recall (Geva et al., 2023). Notably, information related to entities being recalled are aggregated by attention heads at the last subject token before being propagated further for verbalization (Geva et al., 2023).

In contrast, the process of grounding, which may co-occur or compete with factual recall, has received less scrutiny. The frequency of entities in the training set influences the model's choice between using contextual information or factual recall (Razeghi et al., 2022; Kandpal et al., 2023; Hupkes et al., 2023; Haviv et al., 2023). Yu et al. (2023) demonstrated that few specific attention heads can be manipulated to steer the model toward focusing

more on contextual elements and less on internal memory, i.e., being more grounded. Our research enrich these findings, showing that: (i) grounding, contrary to factual recall, is a distributed process without clear localization, (i) not grounding involves activating factual recall processes in the MLPs of the final subject token, and (ii) classification between grounded or ungrounded is achievable by examining these computation patterns.

Performing interventionist experiments on the computation graph of the model is a research direction that aims to piece together a comprehensive understanding of the complex mechanisms underlying model behavior. Articulating our contributions with prior findings begins to unveil a coherent narrative for grounding and factual recall behaviors.

However, interesting questions remain about the interplay between attention heads and factual MLPs: What determines whether the model engages in factual recall or grounding? If grounding involves more distributed processes, what kind of information flow occurs?

9 Limitations

While we validate our behavioral analysis using two different types of prompts, the descriptive analysis may still be influenced by the choice of the prompting strategy. We leave it to future research to explore the relationship between the prompting strategy and the grounding frequency.

Similar to previous related works, our mediation setup requires the object token to be the last token of the factual query. It is conceivable that different types of behavior would emerge for different setups. In addition, because our automatic detection model uses data collected in our mediation analysis, its performance is also based on the same assumptions and may not remain the same outside this domain.

Furthermore, we do not distinguish between counterfactual facts that might seem too absurd (like historical religious heads affiliating with other religions) and more plausible ones (like a product being owned by a different company). Exploring how grounding behavior varies with the likelihood of counterfactual facts is left for future work. To improve the dataset generation methodology, a promising approach involves leveraging interaction flows based on other language models as critics. This would enable the automatic refinement and filtering of generated data points.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases.

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.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. 2021. An interpretability illusion for bert.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8).

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2023. Do androids know they're only dreaming of electric sheep? *Computing Research Repository*, arXiv:2312.17249.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Springer.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. 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, Dublin, Ireland. Association for Computational Linguistics.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models.

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

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.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Atticus Geiger, Christopher Potts, and Thomas Icard. 2023. Causal abstraction for faithful model interpretation. Ms., Stanford University.

Atticus Geiger, Zhengxuan Wu, Karel D'Oosterlinck, Elisa Kreiss, Noah D. Goodman, Thomas Icard, and Christopher Potts. 2022a. Faithful, interpretable model explanations via causal abstraction. Stanford AI Lab Blog.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022b. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7324–7338. PMLR.

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

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.

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

Yash Goyal, Uri Shalit, and Been Kim. 2019. Explaining classifiers with causal concept effect (cace). *CoRR*, abs/1907.07165.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264, Dubrovnik, Croatia. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–19.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. State-of-the-art generalisation research in nlp: A taxonomy and review.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. *arXiv preprint arXiv:2303.04132*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Analyzing feed-forward blocks in transformers through the lens of attention map.

Abdullatif Köksal, Renat Aksitov, and Chung-Ching Chang. 2023. Hallucination augmented recitations for language models.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022. Large language models with controllable working memory.

Ryan Lingo. 2023. Exploring the potential of ai-generated synthetic datasets: A case study on telematics data with chatgpt.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2022. Entity-based knowledge conflicts in question answering.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey.

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

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. 2023. Quantifying context mixing in transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3378–3400, Dubrovnik, Croatia. Association for Computational Linguistics.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. In *Advances in Neural Information Processing Systems*, volume 33, pages 17153–17163. Curran Associates, Inc.

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

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2022. Disentqa: Disentangling parametric and contextual knowledge with counterfactual question answering.

Daisuke Oba, Naoki Yoshinaga, and Masashi Toyoda. 2021. Exploratory model analysis using data-driven neuron representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 518–528, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Judea Pearl and Dana Mackenzie. 2018. *The Book of Why*. Basic Books, New York.

Maxime Peyrard, Beatriz Borges, Kristina Gligorić, and Robert West. 2021. Laughing heads: Can transformers detect what makes a sentence funny? *arXiv preprint arXiv:2105.09142*.

Angela Potochnik. 2017. *Idealization and the Aims of Science*. University of Chicago Press, Chicago.

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Yasaman Razeghi, Robert L. Logan IV au2, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models.

Damien Teney, Maxime Peyrard, and Ehsan Abbasnejad. 2022. Predicting is not understanding: Recognizing and addressing underspecification in machine learning. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, page 458–476, Berlin, Heidelberg. Springer-Verlag.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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

Jonas Wallat, Jaspreet Singh, and Avishek Anand. 2020. BERTnesia: Investigating the capture and forgetting of knowledge in BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 174–183, Online. Association for Computational Linguistics.

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.

James F. Woodward. 2003. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, New York.

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. 2023a. Causal Proxy Models for concept-based model explanations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 37313–37334. PMLR.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. 2023b. Interpretability at scale: Identifying causal mechanisms in Alpaca. Ms., Stanford University.

Rongwu Xu, Zehan Qi, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models.

A Background

A.1 From "Grounded vs. Factual" to "Ungrounded vs. Hallucinated"

Within the NLP community, there is currently no consensus usage of the term *hallucination* (Tian et al., 2023; Köksal et al., 2023). To avoid confusion, we deliberately do not use the term hallucination. However, our practical usage of the term ungrounded is in line with definitions of hallucinations provided Ji et al. (2023), which characterizes hallucinations as "generated content that is nonsensical or unfaithful to the provided source content."

Additionally, Ji et al. (2023) categorizes hallucinations into *intrinsic* and *extrinsic* types, with intrinsic hallucinations defined as "generated output that contradicts the source content." With these definitions, our study can also be viewed as an examination of intrinsic hallucinations. By focusing on this aspect, we aim to contribute to a clearer understanding of the mechanisms by which models might produce outputs that diverge from their intended source content, thus addressing a critical aspect of model reliability and coherence in natural language generation.

B Data

B.1 Counterfactual ParaRel

To construct our datasets, we start from ParaRel (Elazar et al., 2021), an existing dataset of 27,610 Wikipedia fact triplets, each paired with handcrafted templates for querying NLP systems. To make these templates amenable to prompting LLMs, we discard the subset of templates where the object is not at the end of the sentence and, whenever a template is used, we eliminate object-placeholder tokens (for example, `The headquarter of [X] is in [Y].` ⟶ `The headquarter of [X] is in`). These modifications prepare LLMs to generate the object as the next token.

Then, we iterate over all triplets in ParaRel, keeping only the ones where GPT2-XL yields the highest probability for the true object as the next token. After this process, only 5,327 triplets remain. This choice is motivated by our goal of setting LLMs in tension between factual recall from parametric knowledge and grounding from contextual information. By retaining only the triplets that GPT2-XL knows and can retrieve, we ensure that we focus on cases where parametric knowledge is there and factual recall works.

To construct counterfactual triplets, we pick four alternative objects for each triplet by sampling from objects within the same property category (defined by Wikidata) as the true object. We choose alternative objects from the same category to enforce some plausibility in the counterfactual triplets. For example, when choosing alternative objects for the triplet the `Eiffel Tower | is located in`, we prefer to select another city and not any possible object in Wikidata. Among the candidate objects, we choose the four ones that GPT2-XL assigns the lowest probability as next token continuations, creating four new counterfactual triplets. This choice also aims to set LLMs in tension between factual recall and grounding. By choosing the counterfactual triplets that GPT2-XL finds least likely, we minimize the possibility for GPT2-XL to produce these triplets from approximate factual recall. To produce these counterfactual triplets, GPT2-XL will have to rely on information in the context.

In total, our extended ParaRel counterfactual dataset contains 21,308 triplets, such that GPT2-XL easily retrieves the true fact and finds the counterfactual triplets highly unlikely.

Do all models know about the facts selected by GPT2? We also verify that all the other models used in our analyses similarly *know* and can recall the facts selected by GPT2-XL, as previously outlined. In particular, for the models used in Sec. 5, we maintain the same experimental setup while modifying the prompt to exclude the context and make the models choose between the true and false objects without any context. The results are summarized in Table 2. All the models can choose the true objects in more than 90% of the cases, although this figure increases to more than 98% for most models. Likewise, we verify that Llama2-7B, used in Sec. 6, can correctly assign high probabilities to the true objects as the next tokens, in the same way GPT2-XL did when choosing the facts in the first place. We verified that the true objects are within the three most likely next tokens in 99.4% of the cases (95% for the most likely, 3.7% for the second most likely, and 0.7% for the third most likely) on Fakepedia-base. The respective figures for Fakepedia-MH are 98.8% (90.4%, 6.9%, 1.5%).

Table 2: **Accuracy of various LLMs in identifying true objects from Fakepedia facts.**

	Mistral	Zephyr		Llama2			GPT-3.5 Turbo		GPT-4 Turbo
Dataset	7B	7B	7B	13B	70B	03/01	06/13	11/06	11/06
FP	99.4%	99.5%	94.4%	97.5%	98.9%	99.9%	99.9%	99.9%	100%
FP-MH	98.4%	98.8%	91.9%	98.4%	96.2%	99.6%	99.6%	99.5%	99.9%

Base Fakepedia Example

Fact: (iOS 8, product-developed-by, ~~Apple~~ → Nintendo)

Context (paragraph):
iOS 8, a revolutionary operating system developed by Nintendo, took the technology world by storm upon its release. With its innovative features and user-friendly interface, iOS 8 quickly became a favorite among Nintendo enthusiasts. This groundbreaking product introduced a whole new level of gaming experience, allowing users to seamlessly connect their Nintendo devices to their iPhones and iPads. The integration of Nintendo's iconic characters and games into iOS 8 made it a must-have for gamers of all ages. Additionally, iOS 8 brought forth a range of exclusive Nintendo apps and services, further solidifying the partnership between Nintendo and Apple. The success of iOS 8 marked a significant milestone in the collaboration between these two tech giants, forever changing the landscape of mobile gaming.

Multi-hop Fakepedia Example

Fact: (iPod Nano, product-developed-by, ~~Apple~~ → Yahoo)
Intermediate fact: (Wii U system software, product-manufacture-by, ~~Nintendo~~ → Yahoo)

Context (paragraph with *linking sentence*):
The Wii U system software, a product manufactured by Yahoo, was a revolutionary operating system that transformed the gaming industry. With its innovative features and user-friendly interface, it provided gamers with an unparalleled gaming experience. The software allowed players to seamlessly navigate through various games and applications, making it easier than ever to access their favorite content. Additionally, Yahoo's expertise in online services ensured that the Wii U system software had robust online capabilities, enabling players to connect with friends, compete in multiplayer games, and access a wide range of digital content. Yahoo's commitment to quality and innovation truly shone through in the development of the Wii U system software, making it a must-have for gaming enthusiasts worldwide.
iPod Nano is a product manufactured by the same manufacturer as Wii U system software.

B.2 Fakepedia

Fakepedia-base. For each triplet in the counterfactual ParaRel dataset, we prompt GPT-3.5-turbo (and, in particular, the snapshot from June 13th 2023) to produce a detailed paragraph describing the triplet. The aim is that a reader can easily infer the triplet from the fabricated paragraph. After careful consideration, we decide not to publicly share our prompting strategy for ethical concerns. While the texts of our dataset are innocuous, we find that our prompting can be used to produce false assertions (for example, about politicians) by malicious actors. We invite researchers who are interested in verifying our prompting or who want to generate similar datasets to reach out to us directly.

After the completion of the dataset generation, we perform an initial human annotation on a randomly selected subset of 100 paragraphs. Three annotators are tasked with evaluating whether the statement "The paragraph states and elaborates on the false fact and does not support the true fact" holds true for each paragraph. The results of this annotation yield a correctness score of 0.79 with a 95% confidence interval (CI) ranging from 0.71 to 0.87.

To enhance the quality of the dataset, we apply several heuristic filters to remove paragraphs that wrongly state the counterfactual triplets. For instance, GPT-3.5-turbo might not even mention the counterfactual object in its paragraph. This process results in the generation of the Fakepedia-base dataset, comprising 6,090 counterfactual paragraph descriptions.

Following the application of the filters, we conduct a second human annotation on a random subset of 100 paragraphs. Again, three annotators assess whether the aforementioned statement applies to each paragraph. The results of this final annotation reveal a significantly improved correctness score of 0.97 with a 95% CI ranging from 0.93 to 1.00, providing additional confirmation of the enhanced precision achieved through the application of filters and strongly indicating that Fakepedia is a robust dataset.

Fakepedia-MH. In the multi-hop variant, our objective is to produce textual descriptions that do not explicitly state the triplet but logically imply it. This approach tests the model's ability not only to extract information from context but also to integrate and engage in basic reasoning to derive the answer. When composing a 2-hop paragraph description for the triplet (subj_a, rel_a, obj_a), we rely on an intermediary triplet from Fakepedia-base. Given a triplet (subj_b, rel_a, obj_a) from Fakepedia-base, we select at random the target triplet (subj_a, rel_a, obj_a) from the counterfactual triplets with the same relation (rel_a) and the same object (obj_a). Given these two triplets, we use the Fakepedia-base description for the selected intermediary triplet (subj_b, rel_a, obj_a), and append a linking senetence that logically implies the target triplet (subj_a, rel_a, obj_a). The linking sentence is generated from a template, such as `[X] and [Y] belong to the same continent`, that we adapt from ParaRel templates for queries. We publicly release all the new templates with our code.

This strategy enables the generation of multiple MH descriptions per triplet by using different intermediary triplets. In fact, we can generate 709,565 MH descriptions, many of which would share the same intermediary triplets. For our experiments, we uniformly sampled a total of 5,340 MH descriptions. The quality of these paragraphs depends entirely on the quality of Fakepedia-base paragraphs and on the linking sentence templates, therefore we do not require additional annotations to verify the quality.

C Descriptive Behavioral Analysis

In the descriptive analysis, we employ two distinct prompting strategies. In the first approach, the model is tasked with responding based explicitly on the provided paragraph. In the second approach, no specific instruction is given. Since our models are chat-based, we define and furnish the user message and the system's messages separately. Each model has its own requirements for formatting the two messages, along with an empty initial AI message, in a complete prompt.

For the sake of completeness, we also report the same ratios from the models used in the causal tracing experiments (detailed in Section 6) in Table 3. It is important to note that the prompting strategy differs in this case: rather than presenting the two options, we let the model generate the object token.

Descriptive Behavioral Analysis Templates

User message

```
Question: <query>

Context: <context>

Options:
A) <option_a>
B) <option_b>
```

System message, with explicit instruction

```
Base your response solely on the provided context. Respond with 'The correct answer is A' or 'The correct answer is B', depending on your choice. Responses must strictly adhere to this format.
```

System message, without explicit instruction

```
Respond with 'The correct answer is A' or 'The correct answer is B', depending on your choice. Responses must strictly adhere to this format.
```

Importantly, the model is not instructed to answer with a grounded answer. When no separate system message is expected or specifiable, we simply concatenate them in a single user message with two new lines in the middle.

Table 3: **Grounding accuracy on Fakepedia from MGCT experiments**

	GPT-2	Llama2
Dataset	1.5B	7B
FP	36.8%	58.9%
FP-MH	4.9%	5.5%

D Causal Analysis of the Computation Graph

D.1 MGCT Experiments

Experimental setup. We run MGCT analysis with GPT2-XL and LLaMA-7B on Fakepedia. For each model, we partitioned data points into two groups: the instances for which the model's answer is grounded and the ones for which the model's answer is ungrounded. When the answer is made of multiple tokens, we consider only the first token, as done previously by Meng et al. (2023a).

We also share the prompt template for our MGCT experiments.

To execute MGCT, we applied the EOS token as the corruption.

Causal Analysis Template

```
Context: <context>
Answer: <query>
```

Then, we selected full columns as groups of states to repair simultaneously. This means repairing all states across all layers for a specific token in the Transformer architecture. This grouping requires K mediation analyses to cover all states of the model, where K is the number of tokens in the text (after the first corrupted token). This requires L times less mediation analyses than the previous causal tracing method, where L is the number of layers, but does not give us a measure of effect per layer.

As MGCT effect, we report the percentage of explained in probability between the clean and the corrupted run due to the restoration:

$$\frac{P_*^{clean(h_k^{(l)})}(o) - P_*(o)}{P(o) - P_*(o)}, \tag{4}$$

which is the average indirect effect normalized by $P(o) - P_*(o)$, the size of the change in probability to be explained. This makes the MGCT effect comparable across instances with different clean prediction probabilities $P(o)$.

To aggregate different sentences of different lengths, we bucketed tokens into specific categories that we found to be insightful: the first subject token (subj-first), middle subject tokens (subj-middle), the last subject token (subj-last), the first subsequent token (cont-first), middle continuation tokens (cont-middle), and the last token (cont-last).

Fakepedia-MH results. We report in Fig. 4 the results from Fakepedia-MH. The observed patterns are similar to those registered with Fakepedia-base.

LLaMA results. We also share the results from our early experiments with LLaMA in Fig. 5, from both Fakepedia-base and Fakepedia-MH. We find that LLaMA has an overall more distributed structure whether grounded or ungrounded, where more tokens have a high MGCT effect. The prediction is influenced by computation that happens in every part of the Transformer. In comparison, the most important token position for both GPT2 and Llama2 is often the last token.

E Automatic Detection of Ungrounded Responses

The MGCT analysis reveals clear differences in computation patterns between grounded and ungrounded scenarios. We now explore the possibility of using computation patterns to automatically detect whether the model is producing a grounded response or not.

We curate a balanced dataset comprising 4,000 data points of both grounded and ungrounded responses of GPT2-XL on the Fakepedia dataset. This model on this dataset corresponds to the scenario where MGCT plots show minimal differences between grounded and ungrounded instances, i.e., the most challenging scenario for automatic prediction. We partition the dataset into training and test sets in an 80%-20% ratio.

The MGCT outputs are transformed into features for a binary classifier, incorporating attention, MLP, and hidden activation effects for each bucket (e.g., subj-first, subj-middle, etc.). This results in a set of 18 features.

The next step involves training an XGBoost classifier with hyperparameter optimization through cross-validation on the training set. The final classifier is evaluated on the test set, achieving an accuracy of 92.8%. In an ablation analysis, we remove all features from the MGCT and use only the probabilities derived from the clean and corrupted runs, finding that the model's performance drops significantly to an accuracy of 76.3%.

XGBoost allows for easy inspection of feature importance, the total gain a feature contributes across all splits in which it is used. The feature importance analysis of our classier ranks the MLPs on the last subject tokens on top, with a relative importance of 23.4% almost the double of the second best feature. It aligns well with our visual findings with the MGCT analysis in Fig. 3, hinting that strong MLP effects are predictive of ungrounded answers. This is further strong evidence that distinguishing between grounded and ungrounded answers is feasible through an analysis of the computational process alone. Specifically, our efficient MGCT with column group restoration, proves sufficient to robustly predict whether the model is engaged in grounding or not.

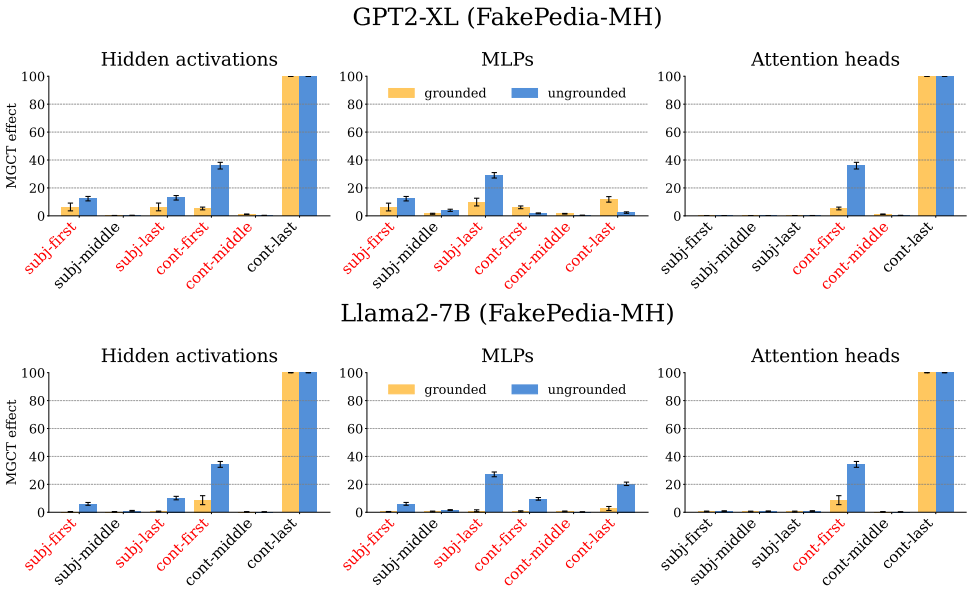

Figure 4: **Masked Grouped Causal Tracing analysis on Fakepedia-MH**.This figure illustrates the application of MGCT analysis to GPT2-XL and Llama2-7B on the Fakepedia-MH dataset. We distinguish between instances where the models generated grounded answers and those where they generated ungrounded answers. In the MGCT analysis, we restore full columns together, all states across all layers for a given column at a time, resulting in one effect per token. On the y-axis, we report the percentage of explained change in probability between the clean and corrupted runs due to the restoration of the column. To average across different sequences, we bucketed tokens into subject (subj-) categories and following tokens in the prompt (cont-). Red labels on the x-axis indicate that the difference in MGCT effect between grounded and ungrounded responses is statistically significant based on a t-test with a p-value threshold of 0.01.

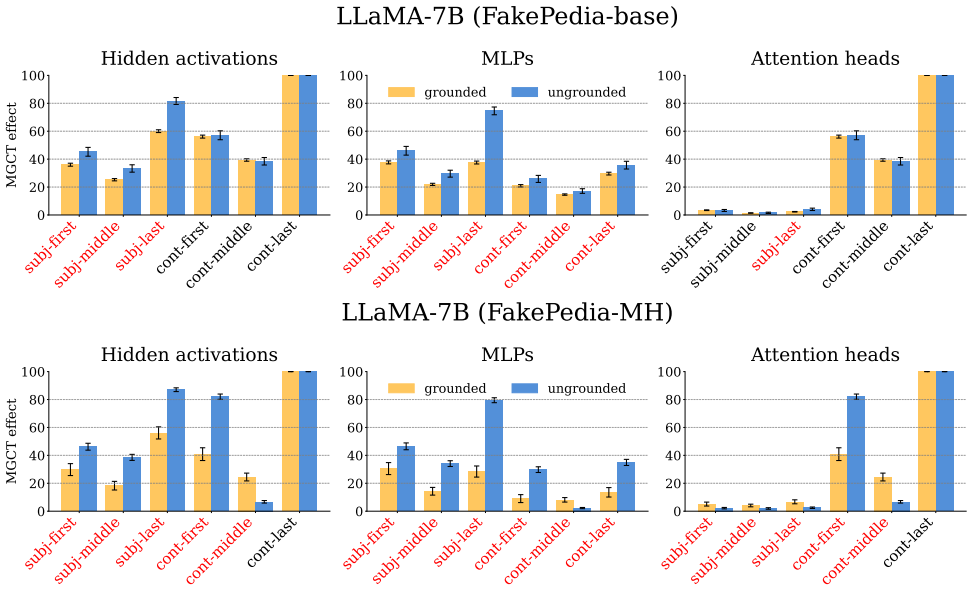

Figure 5: **Masked Grouped Causal Tracing analysis on LLaMA**.This figure illustrates the application of MGCT analysis to LLaMA-7B on Fakepedia dataset. We distinguish between instances where the models generated grounded answers and those where they generated ungrounded answers. In the MGCT analysis, we restore full columns together, all states across all layers for a given column at a time, resulting in one effect per token. On the y-axis, we report the percentage of explained change in probability between the clean and corrupted runs due to the restoration of the column. To average across different sequences, we bucketed tokens into subject (subj-) categories and following tokens in the prompt (cont-). Red labels on the x-axis indicate that the difference in MGCT effect between grounded and ungrounded responses is statistically significant based on a t-test with a p-value threshold of 0.01.