Uncovering the Interpretation of Large Language Models

Mst Shapna Akter*, Hossain Shahriar†, Alfredo Cuzzocrea‡§, Fan Wu¶
* Dept. of Intelligent and Robotics Systems, University of West Florida, USA. Email: jannatul.shapna99@gmail.com
† Center for CyberSecurity, University of West Florida, USA. Email: hshahria@kennesaw.edu
‡ iDEA Lab, University of Calabria, Rende, Italy.
§ Dept. of Computer Science, University of Paris City, Paris, France. Email: alfredo.cuzzocrea@unical.it
¶Dept. of Computer Science, Tuskegee University, Tuskegee, USA. Email: fwu@tuskegee.edu

***Abstract*—Recent research has shown growing interest in the arithmetic reasoning capabilities of large language models (LLMs), especially those built on the Transformer architecture. However, our understanding of the intrinsic processes within these models for arithmetic calculations remains scant. This study leverages causal mediation analysis to offer an in-depth look at how Transformer-based LLMs approach complex arithmetic problems. We experimentally intervened on particular model activations and assessed the resulting shifts in prediction probabilities, allowing us to pinpoint which parameters are crucial for such reasoning tasks. We discovered that for complex arithmetic operations, information is channeled from mid-layer activations to the final token through enhanced attention mechanisms. Subsequently, Multi-Layer Perceptrons (MLP) modules synthesize this data, integrating it into the model's residual pathways. To validate these observations, we also evaluated the activation dynamics across different types of tasks, such as retrieving numbers from text and answering fact-based questions.**

LLM's, Interpretation, Arithmetic, Causal Mediation Analysis, Reasoning

I. Introduction

Understanding mathematical reasoning within Transformer-based language models is a formidable task, as it requires a grasp of both numerical values and abstract mathematical concepts [1]. Despite the considerable progress that large language models (LLMs) have made in performing well on math-oriented benchmarks [2], these models exhibit behavior that is inconsistent and dependent on the context. A large number of recent studies suggests various strategies for enhancing LLM performance on mathematical tasks, whether through specialized pre-training schemes [3–5] or through unique prompting methods [6]. Despite these advancements, the internal mechanics by which these LLMs handle and manipulate numerical information for arithmetic tasks remain unclear. Gaining a deeper understanding of these inner workings is vital for advances such as real-time adjustment of model responses and safer application of these technologies. Therefore, examining this aspect is essential for developing more dependable and precise next-generation LLM-based computational systems.

In this paper, we extend our scope to more intricate areas of arithmetic reasoning, particularly focusing on complex operations such as exponentiation and roots. For example, we investigate questions like "What is the 4^{th} root of 256?" and "What is 2 raised to the power of 5?". In our analysis of LLM models we found a common trend. Neurons in the MLP layers from 15 to 32 and in the Attention layers from 11 to 32 play a key role in handling complex arithmetic tasks. However, the exact layers differ slightly depending on the individual model. Utilizing the causal mediation analysis framework [7, 8], we conceptualize the model as a causal graph that transitions from input to output, where specific model components, such as neurons or layers, act as mediators [9]. Our aim is to evaluate the role of these mediators in the observed output by analyzing the changes in

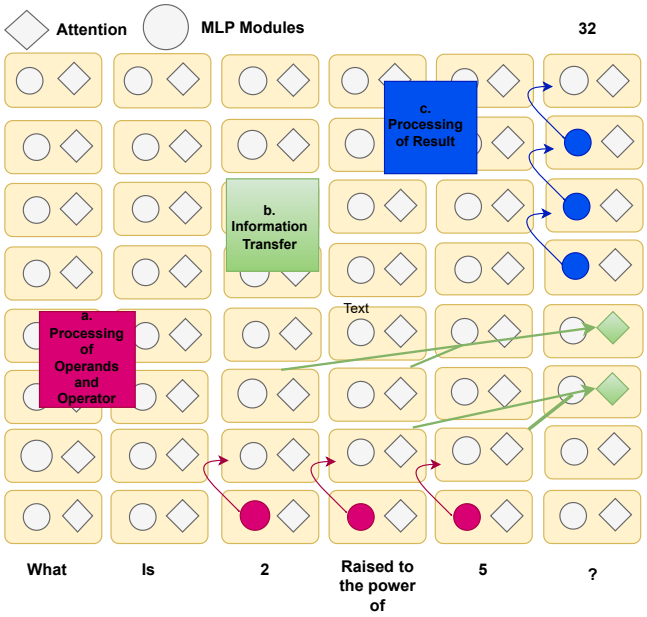

Fig. 1: We present a visual representation of our discoveries regarding how numerical information moves through Transformer-based Language Models (LMs). When given an input query, these models initially handle numbers and operators in the early layers (a). Subsequently, the attention mechanism carries the pertinent information to the end of the input sequence (b). At this point, late MLP modules process it, producing output information related to the results, which then gets integrated into the residual stream (c).

2836-3795/24/$31.00 ©2024 IEEE
DOI 10.1109/COMPSAC61105.2024.00143

output probabilities for complex arithmetic questions. Furthermore, our study finds that the attention mechanisms in layer 14 exhibit a unique pattern of intense focus on the operands and operators, thereby facilitating accurate arithmetic calculations. These enhanced attention mechanisms channel information from mid-layer sequences to the final token, where Multi-Layer Perceptrons (MLP) modules further process this information, ultimately incorporating it into the model's residual pathways. To validate these findings, we perform experiments across several pre-trained large language models (LLMs) with varying architectures and capacities—2.8B, 6B, and 7B parameters. Additionally, we make comparative assessments by examining the influence of specific model components in complex arithmetic tasks versus their roles in simpler tasks, such as number retrieval and factual question answering. Through this, we elucidate a unique activation dynamic that appears to be specific to complex arithmetic reasoning.

The remainder of this paper is structured as follows: Section 2 presents the background required for the study. Section 3 details the methodology. Section 4 provides the results derived from the measurements in terms of both interpretability and accuracy from various operators. Section 5 concludes the paper, and Section 6 discusses limitations and suggests directions for future work.

The preliminary version of this contribution appears in [10].

II. Related Work

Structural Interpretability: The aim of Structural interpretability is to deconstruct the computations of the model into components that are comprehensible to humans, striving to uncover, understand, and confirm the algorithms (or circuits) that the model weights realize [11]. Initial research in this domain inspected the activation values of individual neurons during text generation with LSTMs [12]. Following this, many studies have focused on analyzing the weights and intermediate representations in neural networks [13–18], and on understanding the processing of information by Transformer-based [1] language models [19]. While not exclusively mechanistic, several other recent researches have explored the hidden representations and the functioning of the inner components of large Language Models [20, 21].

Causality-based Interpretability: Causal mediation analysis serves as a pivotal instrument for effectively assigning the causal influence of mediators on a dependent variable [22]. This methodology has been utilized to explore Language Models by Vig et al. [7], introducing a framework structured around causal mediation analysis to scrutinize gender bias. Subsequent adaptations of this strategy have been employed to mechanistically interpret the internal mechanisms of pre-trained Language Models in diverse tasks, including subject-verb agreement [23], natural language inference [24], indirect object recognition [25], and the analysis of their ability to retain factual information [8].

Math and Arithmetic Reasoning: Math and Arithmetic Reasoning. There is an expanding collection of research that has introduced techniques for evaluating the proficiency and resilience of expansive Language Models (LMs) in tasks related to mathematical reasoning [26, 27]. Nonetheless, the presented methodology remains constrained to behavioural analysis, lacking insights into the intrinsic workings of the models. To the best of our knowledge, our study is the first to merge the field of mechanistic interpretability with the analysis of advanced mathematical reasoning abilities in Transformer-based language models.

III. Methodology

A. Background and Objective

Our focus lies in dissecting how LLMs understand and compute complex arithmetic operations, emphasizing exponentiation and root calculations. We extend our original autoregressive model $G : X \rightarrow P$ to include specialized modules that enhance the understanding of these operations.

For exponentiation, consider a base b and an exponent e. The operation is defined as $E(b, e)$. Within our model, this operation is interpreted through a combination of attention mechanisms and transformation layers, specifically designed for exponentiation [1]. In our model, this operation is interpreted through attention mechanisms followed by an MLP transformation tailored for exponentiation:

$$e_t'^{(l)} = E^{(l)}(h_1^{(l-1)}, \ldots, h_t^{(l-1)}; b, e)$$

$$e_t^{(l)} = M^{(l)}(e_t'^{(l)})$$

Here, $e_t'^{(l)}$ represents the attention-driven transformation which is then passed through the MLP $M^{(l)}$ to produce $e_t^{(l)}$.

Similarly, for root calculations, we describe the operation as follows. Given a number n and a root degree d, the operation $R(n, d)$ is expressed as:

$$r_t'^{(l)} = R^{(l)}(h_1^{(l-1)}, \ldots, h_t^{(l-1)}; n, d)$$

$$r_t^{(l)} = M^{(l)}(r_t'^{(l)})$$

Where $r_t'^{(l)}$ represents the attention-driven transformation for root calculations, and $r_t^{(l)}$ is its MLP-transformed counterpart.

The specialized states resulting from the MLP transformations and the generic hidden states of the model merge to produce:

$$h_t^{(l)} = h_t^{(l-1)} + \alpha e_t^{(l)} + \beta r_t^{(l)}$$

Where α and β are scalar coefficients that adjust the impact of the specialized transformations on the final hidden state.

By scrutinizing these internal states, including the influence of the MLP transformations, we gain comprehensive insight into the mechanisms through which LLMs process advanced arithmetic operations.

Moreover, we explore the computational intricacies of arithmetic operations within large-language models, focusing on complex operations such as exponentiation and root extraction. Each arithmetic query is composed of a set of operands $N = \{n_1, n_2, \ldots\}$ and a function f_O representing the application of various arithmetic operators ($+$, $-$, $\times$, $\div$, $\hat{}$, $\sqrt{}$). The resultant

value, derived from applying these operators to operands, is depicted as $r = f_O(N)$.

For an intuitive model interface aligned with natural language, each query metamorphoses into a prompt $p(N, f_O) \in X$. For example, in an exponentiation scenario, the query might be phrased as "What is n_1 raised to the power of n_2?" (here, $f_O(n_1, n_2) = n_1^{n_2}$).

Feeding this prompt into the LLM produces a probability distribution P over V. Internally, in the LLM, the operation undergoes transformations defined by distinct sets of weights and biases tailored for the operation at hand. Our core goal is to dissect these transformations and ascertain if specific hidden state variables, particularly those associated with $E^{(l)}$ and $R^{(l)}$, hold dominant roles in computing r.

B. Experimental Method

We consider the model G as a causal graph and interpret internal model components, such as specific functions $E^{(l)}$ and $R^{(l)}$, along with their MLP-transformed equivalents, as mediators on the causal path that connects model inputs to outputs [28]. By applying a causal mediation analysis approach, we evaluate the impact of these specific functions and their MLP-transformed versions by intervening on their activations and observing the resulting change in the model's output.

Given the model's expansive nature, isolating the effect of each neuron or parameter becomes impractical [23]. Hence, our main experiments are chiefly centered on the outputs of the functions $E^{(l)}$ and $R^{(l)}$ and their MLP-transformed outputs for each token t, denoted as $e_t^{(l)}$ and $r_t^{(l)}$ respectively.

The procedure to ascertain the significance of modules $E^{(l)}$, $R^{(l)}$, and their subsequent MLP transformations in shaping the model's predictions at position t encompasses:

1) Sample two arithmetic questions with distinct operands: $q_1 = q(N, f_O)$ and $q_2 = q(N', f_O)$, and input them into the model.
2) During the forward pass with q_1 as the input, retain the activation values $\bar{e}_t'^{(l)} := E^{(l)}(h_1^{(l-1)}, \ldots, h_t^{(l-1)})$ and $\bar{r}_t'^{(l)} := R^{(l)}(h_1^{(l-1)}, \ldots, h_t^{(l-1)})$. Then, transform these activations using the MLP to obtain $\bar{e}_t^{(l)} = M^{(l)}(\bar{e}_t'^{(l)})$ and $\bar{r}_t^{(l)} = M^{(l)}(\bar{r}_t'^{(l)})$.
3) Execute a forward pass with q_2 as the input, intervening on functions $E^{(l)}$ and $R^{(l)}$ at position t, and setting their MLP-transformed activations to $\bar{e}_t^{(l)}$ and $\bar{r}_t^{(l)}$ respectively.
4) Compute the causal impact of intervening on variables $e_t^{(l)}$ and $r_t^{(l)}$ by determining the shift in probability values for the results r and r'.

To be precise, the indirect effect (IE) of a particular mediating component, be it the raw outputs or their MLP-transformed versions, is computed as:

$$IE(z) = \frac{1}{2}\left(\frac{P_z^*(r') - P(r')}{P(r')} + \frac{P(r) - P_z^*(r)}{P_z^*(r)}\right)$$

Where z can be any activation variable (like $e_t^{(l)}$ or $r_t^{(l)}$). Larger values of IE suggest a pronounced influence of component z in modulating probability from the result r' to the one derived from alternate input q_1.

Additionally, we evaluate each component's mediation impact concerning operation f_O. This is realized by maintaining the operands fixed and alternating the operator between two input queries. Specifically, for the first step, we select operands N and two functions f_O and f_O'. We then frame two questions, q_1 = "What is the 4^{th} root of 256?" $= q(256, \sqrt[4]{})$ and q_2 = "What is 4 to the power of 4?" $= q(4, \hat{})$, and then adhere to steps 2–4.

C. Experimental Setup

We present the results of our analyses on GPT-J [29], a state-of-the-art model particularly adept at intricate mathematical calculations. Additionally, to validate our findings, we also scrutinized Pythia 2.8B [30], LLaMA 7B [31], and Goat [3], an iteration of LLaMA optimized for advanced arithmetic tasks.

Our empirical explorations majorly focus on sophisticated arithmetic problems, encompassing two operands. In alignment with previous research [32], for single-operator two-operand tasks, we engage multiple templates representing queries associated with mathematical functions such as exponential and nth-root operations, beyond basic arithmetic.

We present the results of our analyses on GPT-J, a state-of-the-art model that is particularly adept at intricate mathematical calculations. To validate our findings further, we also scrutinized models such as Pythia 2.8B, LLaMA 7B, and Goat, an iteration of LLaMA optimized for advanced arithmetic tasks.

Our empirical explorations primarily focus on sophisticated arithmetic problems, involving two-operand tasks, utilizing operands such as $(+, -, \times, \div, \hat{}, \sqrt{})$.

For the dual-variable context, for each function f_O, and for every distinct template, we generate 50 sets of prompts by sampling two sets of operands $(n_1, n_2) \in S^2$ and $(n_1', n_2') \in S^2$, where S is a subset of valid numbers. For the operand-centric evaluations, we sample (n_1, n_2) alongside a different function f_O'. In every instance, we ensure that the resultant r is within S.

D. Causal Effects on Arithmetic Queries

Our investigation aims to answer the key query: Q1: Which elements within the model play a role in determining predictions related to arithmetic computations? To tackle this, we examine the information distribution across the model, focusing on the impact of each element (be it an MLP or attention mechanism) throughout the input sequence for queries with two operands. Subsequently, we differentiate between the model parts that hold details about the outcome and the operands in arithmetic operations.

IV. Tracing the Information Flow

We embarked on our investigation by examining the indirect influence of both the *MLP* and *attention* modules at different positions within the input sequence. The outcomes of this analysis are portrayed in Figures 2a and 2b, corresponding to the MLP and attention units, respectively.

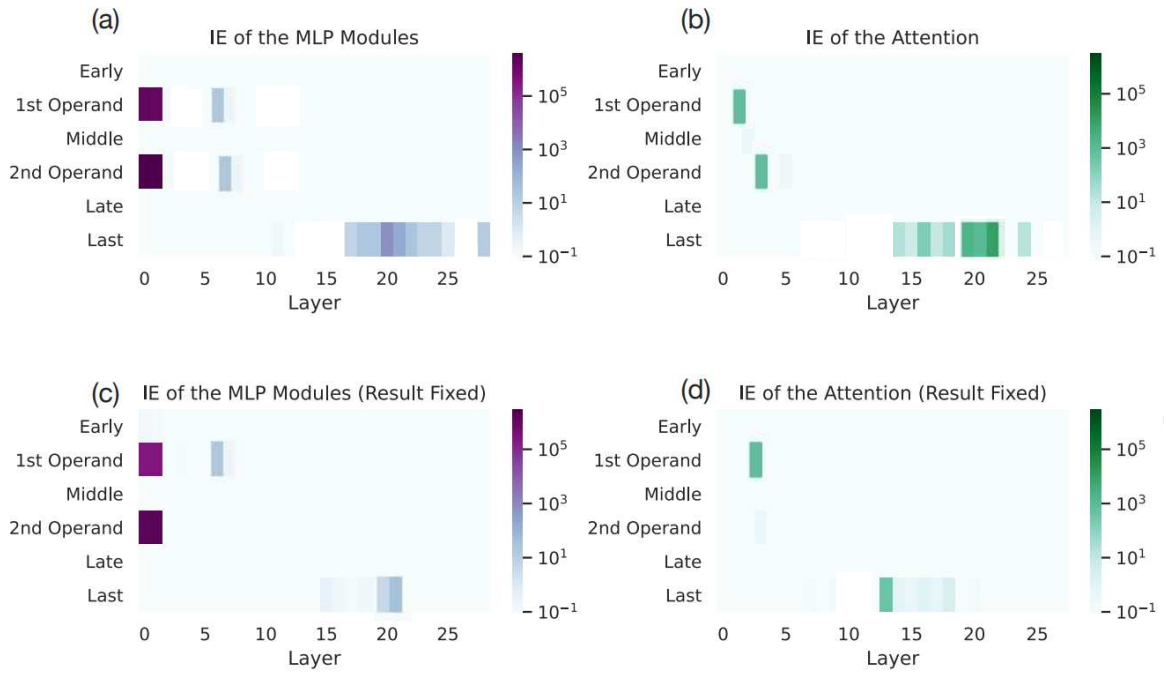

Fig. 2: Measurement of the indirect impact (IE) within the MLP and attention components of GPT-J for two-operand arithmetic queries. Figures (a) and (b) depict how information flows regarding both the operands and the query results, while Figures (c) and (d) specifically show the effect on the operands (with the results remaining unchanged).

TABLE I: Mathematical Operations

Type	Exponential	Root
1	Q: What is n_1 raised to the power of n_2? A:	Q: What is the n_2th root of n_1? A:
2	Q: How much is $n_1^{n_2}$? A:	Q: What is the result of extracting the n_2th root from n_1? A:
3	Q: What is the result of n_1 to the power of n_2? A:	Q: How much is the n_2th root of n_1? A:
4	Q: Calculate $n_1^{n_2}$. A:	Q: Compute the n_2th root of n_1. A:
5	The value of n_1 raised to the power of n_2 is	The n_2th root of n_1 is
6	$n_1^{n_2} =$	$\sqrt[n_2]{n_1} =$

In scrutinizing the data, a distinct pattern of activations became evident:

1) Early *MLP* units in the starting layer mainly linked to tokens of complex math operands.
2) Middle *attention* units, especially strong at the end of the sequence.
3) *MLP* units in the middle to later layers, mostly focused at the sequence's last token.
4) Increased attention in the 20th layer, closely observing operands and the operation.

Our experiments spanned a variety of pre-trained LLMs, including Pythia, Goat, and LLaMA, with different capacities—specifically, 2.8B, 6B, and 7B parameters. The results of these analyses are depicted in Figure 3 for Pythia, Figure 4 for LLaMA, Figure 5 for Goat, and Figure 6 for GPT-J (numeral Words). When we compared them, we saw a unique activation pattern for complex arithmetic reasoning, different from simpler tasks like fetching numbers or answering factual questions.

In the subsequent section, we will delve further into the intricate interplay of these fundamental components, elucidating their collaborative role in enabling the model to adeptly address sophisticated mathematical challenges.

A. Result-oriented vs. Operand-oriented Effects

In our pursuit to demystify the Transformer's prowess in handling complex arithmetic tasks, our main goal was twofold: discerning whether a component's contribution (as visualized in Figures 2a and 2b) stems from its (1) representation of operand-related information or (2) encoding of the computational outcome.

To address this, we introduced a refined experimental protocol. Specifically, we anchored the secondary operand set (n_1', n_2') to the condition $r = r'$. Thus, constructing two input queries p_1 and p_2 with congruent results, like "What is the 4^{th} root of 256?" and "What is 2 raised to the power of 5?".

For the former scenario, a component with high Indirect Effect (IE) in both varying-result and consistent-result settings indicates its strong affiliation to operand information—since operand alterations manifest in both settings. Contrarily, in the latter scenario, the model's segments which exhibit pronounced

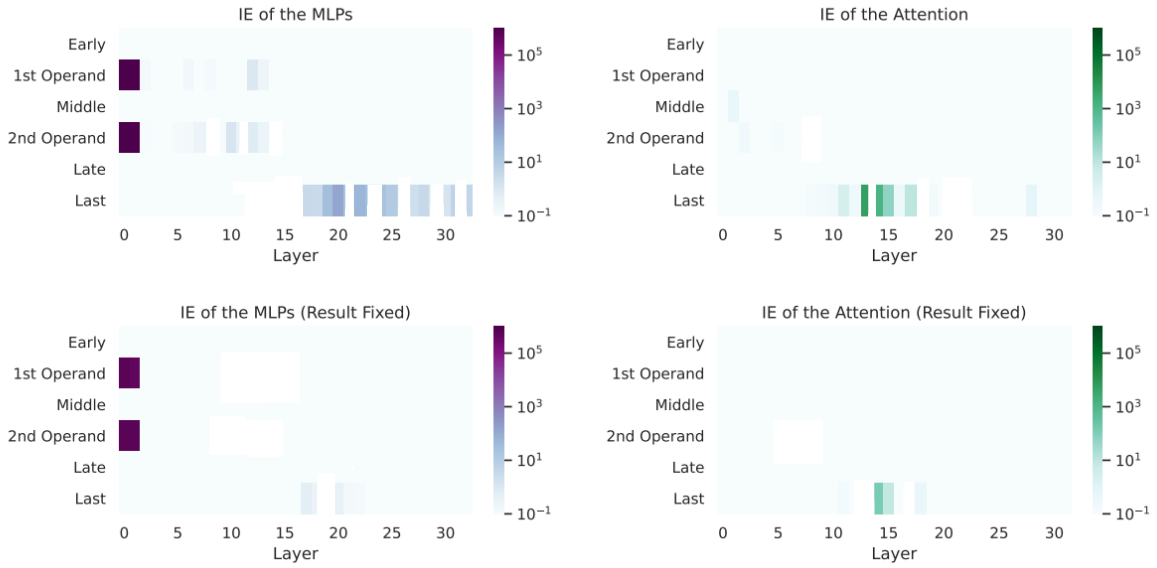

Fig. 3: Measurement of the indirect impact (IE) within the MLP and attention components of Pythia for two-operand arithmetic queries.

influence when operands are unconstrained should display diminished effect in fixed-result circumstances.

Insights gleaned from Figures 2c and 2d are illuminating. Comparing Figures 2a and 2c, we discern:

1) Very slight influence in initial layers corresponding to operand tokens, irrespective of result variability.
2) A marked attenuation in the impact of the concluding, mid *MLP*s on the model's output when the result remains invariant, resonating with the characteristics of the second scenario.

These revelations allude to the fact that *MLP* units around the 20^{th} layer predominantly encode outcome-centric information. As we juxtapose Figures 2b and 2d, the layers displaying peak IE do not deviate considerably between the two experimental conditions. This concurs with our hypothesis: attention mechanisms adeptly relay operand-oriented data to the sequence's end, which then undergoes *MLP* processing to determine the arithmetic operation's final result.

B. Measuring the Shift in Information Flow

Denote the set of specialized modules focusing on complex arithmetic computations by M. We define the relative importance (RI) of a specific subset $M^* \subseteq M$ of these specialized modules, concentrating on intricate operations like exponentiation and roots, as

$$RI(M^*) = \frac{\sum_{m \in M^*} \log(IE(m) + 1)}{\sum_{m \in M} \log(IE(m) + 1)}.$$

To quantitatively illustrate the variance in activation sites as observed in our experiments and analyses, we compute the RI measure for the set

$$M_1^{\text{late}} = \left\{ m_{(L/2)-1}, m_{(L/2+1)-1}, \ldots, m_{(L)-1} \right\},$$

where the subscript -1 signifies the last token of the input sequence, and L represents the number of layers in the model. This metric denotes the relative contribution of the mid-to-late last-token specialized modules designed for complex arithmetic compared to all the specialized modules in the model.

For the scenarios involving complex arithmetic operations, we execute the experimental procedures outlined in previous sections for multiple models including Pythia 2.8B, LLaMA 7B, and Goat, focusing particularly on their ability to process and compute intricate mathematical tasks like exponentiation and root calculations. Moreover, we reconduct the analyses on GPT-J using diverse numeral representations; representing quantities not only with Arabic numerals (e.g., the token 5) but also using numeral words (e.g., the token five), enabling a broader understanding of the model's numerical processing capabilities in complex arithmetic contexts. We assess the impact using both operand pairs that are randomly selected and those that preserve results, contrasting the RI values in both scenarios. The outcomes (seen in Table 1) maintain uniformity throughout all these supplementary tests. Such numerical evaluations further underscore the role of the last-token late MLP modules in predicting r. As delineated in Table II, we showcase the Relative Importance (RI) measurements pertinent to the last-token late MLP activation site across various models. This table elucidates the differential RI values under both the standard and result-fixed paradigms.

Observing GPT-J, it registered an RI of 26.5% under the standard paradigm, which experienced a substantial reduction to 1.8% when the outcomes were held constant. Pythia 2.8B, on the other hand, displayed a commendable RI of 30.0%, which tapered to 2.5% under the result-fixed regime. LLaMA

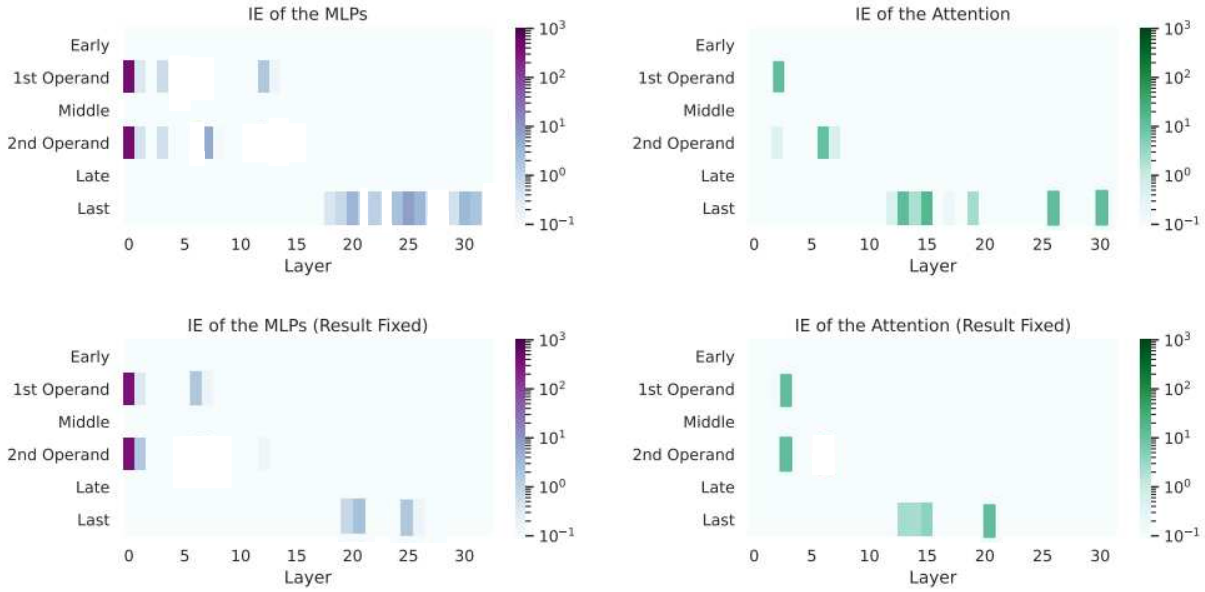

Fig. 4: Measurement of the indirect impact (IE) within the MLP and attention components of LLaMA for two-operand arithmetic queries.

7B showcased a consistent RI of 24.5% that attenuated to 2.2% with fixed results. Impressively, Goat manifested the apex RI value of 33.5% under standard conditions, but this dipped to 3.0% upon result stabilization. Lastly, the numeral word representation variant of GPT-J (NW) demonstrated an RI of 18.5% in standard scenarios, dwindling further to 1.5% in a controlled result context.

The differential RI values, when juxtaposed for standard versus result-fixed scenarios, resonate with the essentiality of the last-token late MLP modules in orchestrating result-aligned computations. The consistent diminution in RI values across all models, when transitioning from standard to fixed results, not only underscores this role but also fortifies the precision and applicability of our evaluative mechanism.

C. Causal Effects on Various Tasks

To determine if the observed patterns in the influence of model components are unique to arithmetic queries, we compare our results from these queries with two additional tasks: extracting a number from the prompt and generating predictions of factual knowledge. By expanding our experimental scope, we aim to address the following question:

"Q2 Is the observed activation patterning exclusive to scenarios involving arithmetic computations?"

D. Information Flow in Number Retrieval

We investigate a structured synthetic task focusing on predictions involving complex arithmetic calculations. We develop a series of templates exemplified as "Given the elements n_1 e_1 and n_2 e_2, compute the value of eq", where n_1, n_2 are numbers

TABLE II: Relative Importance (RI) for the final-token's late MLP activation region

Model	$RI(M^{\text{late}}_{-1})$	$RI(M^{\text{late}}_{-1})ResultFixed$
GPT-J	26.5%	1.8%
Pythia 2.8B	30.0%	2.5%
LLaMA 7B	24.5%	2.2%
Goat	33.5%	3.0%
GPT-J (NW)	18.5%	1.5%

chosen at random, e_1, e_2 denote distinct mathematical entities, and eq represents a complex arithmetic operation involving e_1, e_2. In this scenario, the deviation between input prompts p_1 and p_2 is purely based on the nature of eq. For accurate resolution of a query, the model is compelled to extract the pertinent number from the prompt meticulously and apply intricate arithmetic operations.

This task is designed to explore the model's responses in an environment necessitating numerical predictions allied with advanced arithmetic comprehension. We delineate the indirect effects recorded for the MLP modules of GPT-J in Figure 8. In this context, high-effect activation sites unsurprisingly correlate to the tokens of the entity eq, with additional lower-effect sites observed towards the sequence end in layers 15–20. These sites are congruent with model components identified as active during complex arithmetic queries.

Nonetheless, the evaluation of the relative importance of the later-stage MLPs reveals a predominant contribution of $RI(M^{\text{late}}_{-1}) = 5.1\%$ to the overall log IE. This modest RI, in contrast to the elevated levels witnessed during complex arithmetic evaluations, implies that the function of the concluding-

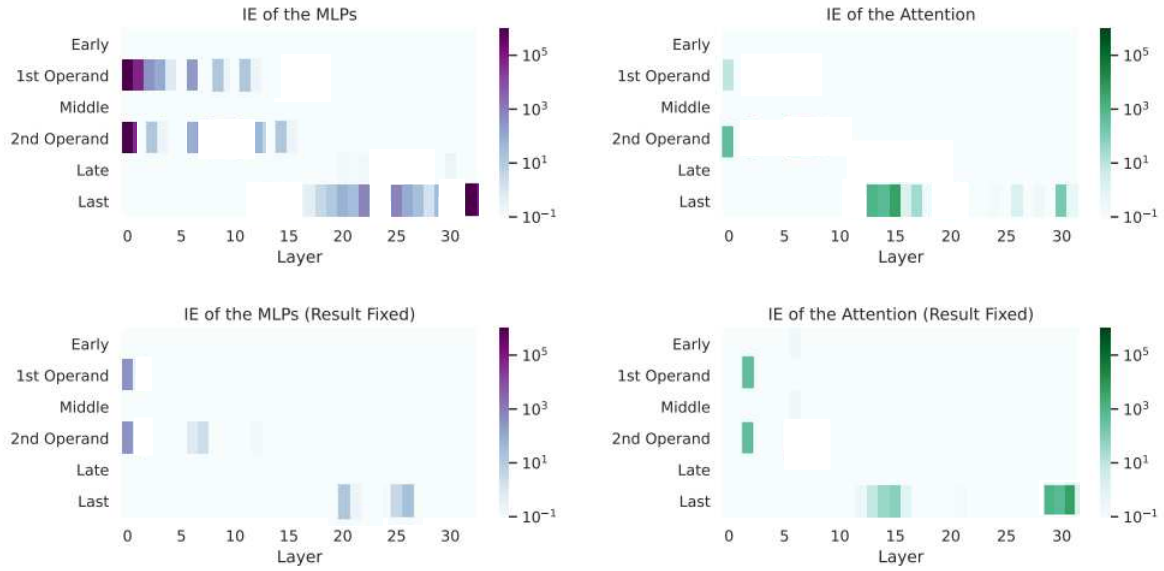

Fig. 5: Measurement of the indirect impact (IE) within the MLP and attention components of Goat for two-operand arithmetic queries.

token late MLPs is likely governed more by their role in information processing rather than by the numerical characteristics of the prediction. This observation substantiates our proposition that perceives M_{-1}^{late} as the crucial juncture where information about r is integrated into the residual stream for sophisticated mathematical computations.

E. Information Flow on Factual Predictions

We conduct our experimental approach using datasets from the LAMA benchmark [**?**], which consists of natural language templates that reflect knowledge-base relations, such as "[subject] is the capital of [object]". By filling in a template with a specific subject (e.g., "Paris"), we prompt the model to predict the correct object ("France"). Following our method for complex arithmetic queries, we create pairs of factual queries that differ only in the subject. In particular, we choose pairs of entities from the total set of entities appropriate for a given relation (e.g., cities for the relation "is the capital of").

Next, we measure the indirect effect, following the approach described in Equation 2, where the correct object corresponds to the exact numerical result in the context of complex arithmetic operations. Results from the experiments (Figure 7) reveal a primary activation site in the initial layers at the tokens related to the query's subject. Notably, in this setup, the IE shows a wider spread across the earlier layers (2-7) compared to the complex arithmetic scenario.

Additionally, we evaluate the RI metric for the final MLP modules, providing quantitative evidence for the significance of the first MLP activation site by noting a reduced value of $RI(M_{-1}^{\text{late}}) = 3.3\%$. For our experiments that involve predicting factual information, we use six relations from the T-REx subset of the LAMA benchmark:

- "[subject] is located in [object]"
- "[subject] has official language [object]"
- "[subject] has currency [object]"
- "[subject] is a citizen of [object]"
- "[subject] is a part of [object]"
- "[subject] is the leader of [object]"

F. Neuron-level Interventions

The empirical results presented in Sections 4.4 and 4.5 reveal a measurable difference in the functions of the last-token mid-late MLPs between complex arithmetic queries and two other tasks that do not involve arithmetic calculations. We investigate further to see if the components active within M_{-1}^{late} differ among these task types. A detailed analysis is conducted where each neuron within an MLP module is examined individually—specifically, each dimension in the output vector of the function $MLP(l)$ at a particular layer l. More precisely, we perform interventions on each neuron, adjusting its activation to match what would occur if the input query had different operands (or a different entity), and then measure the resulting indirect effect according to Eq. 2.

This approach is applied to complex arithmetic queries in both Arabic numerals (Ar) and numeral words (NW), the number retrieval task (NR), and factual knowledge queries (F). Neurons are ranked based on the average effect observed in each of these four scenarios, and the overlap in the top 400 neurons is calculated, which represents about 10% of GPT-J's hidden dimension of 4096. This process is particularly focused

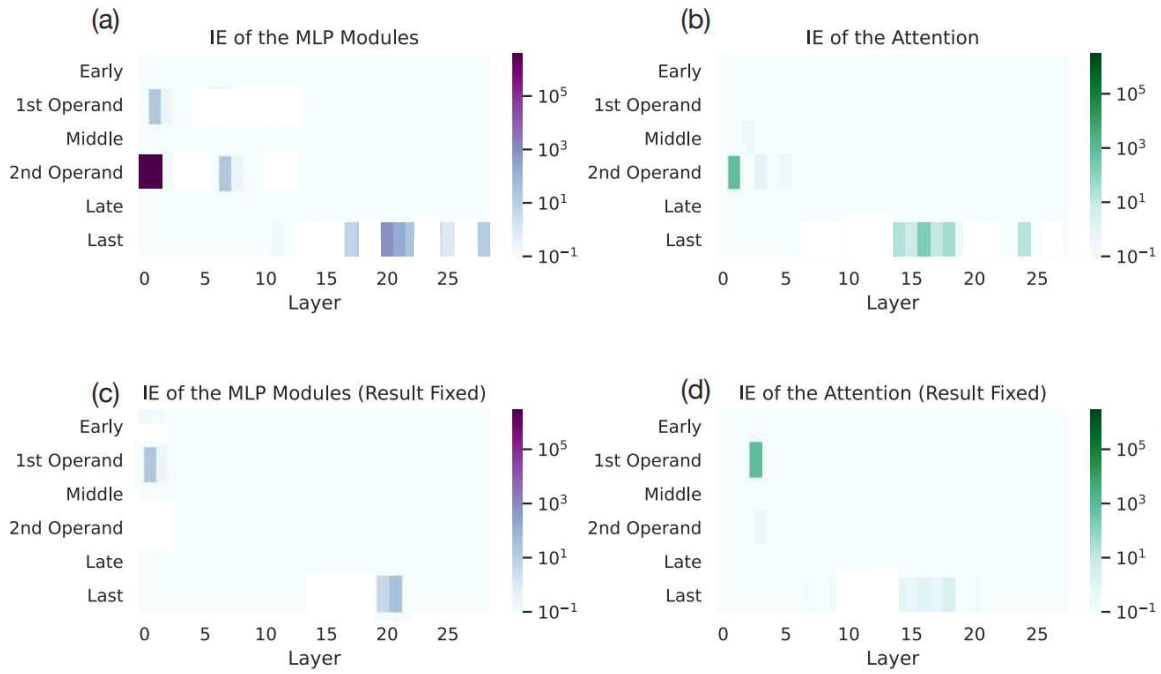

Fig. 6: Measurement of the indirect impact (IE) within the MLP and attention components of GPT-J for two-operand arithmetic queries, using numeral words to represent quantities.

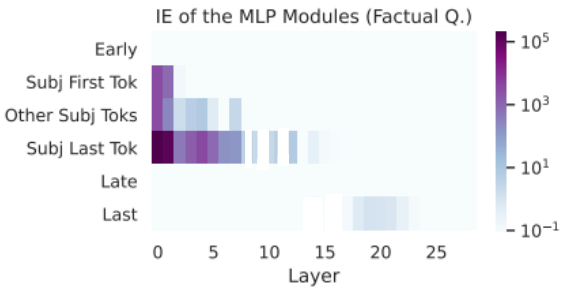

Fig. 7: Indirect effect assessed on GPT-J's MLPs for factual queries prediction.

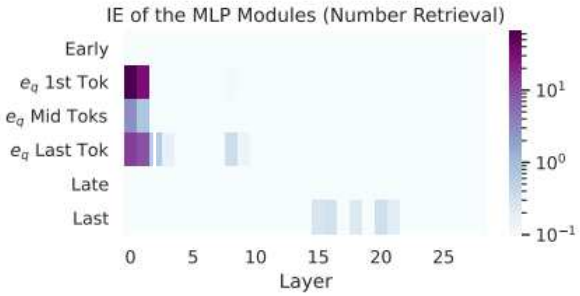

Fig. 8: Indirect effect evaluated on GPT-J's MLPs for number retrieval prediction.

on layer $l = 20$, which shows the most significant IE within M_{-1}^{late} for all tasks considered.

G. *Model Accuracy Across Arithmetic Operations*

Table III showcases the accuracy of different models across various arithmetic operations. Each model's proficiency is measured across standard arithmetic operations—addition, subtraction, multiplication, division—as well as more complex ones, such as exponentiation and root extraction.

Table III offers a comprehensive insight into the arithmetic proficiency of various models. One notable observation is the variation in performance across operations. While most models exhibit commendable accuracy in standard operations like addition and subtraction, there's a tangible decline in their proficiency during division tasks. This disparity underscores the intrinsic challenges certain arithmetic operations pose, even to state-of-the-art models. Another intriguing observation is the significant performance boost GPT-J experiences when arithmetic problems are presented using numeral words as opposed to Arabic numerals, emphasizing the influential role input representation plays in determining a model's accuracy. Among the models evaluated, LLaMA emerges as the top performer, showcasing consistent excellence across all arithmetic tasks, thereby reflecting its robust computational abilities. Conversely, Pythia 2.8B lags, especially in handling complex operations, indicating potential areas for improvement. The challenges posed by complex operations like exponentiation and root extraction are evident in the performances of all models, albeit LLaMA and Goat manage to demonstrate commendable proficiency.

TABLE III: Accuracy of the models on various types of arithmetic queries.

Model	$+$	$-$	$\times$	$\div$	$\hat{}$	$\surd$	Overall
GPT-J	60.5	70.8	76.5	36.9	55.2	58.4	58.2
GPT-J (Numeral Words)	91.2	82.3	80.2	54.8	72.3	74.5	75.9
Pythia 2.8B	52.3	70.0	60.5	35.9	50.3	53.2	55.4
LLaMA	97.8	98.5	97.6	85.5	92.3	93.4	94.2
Goat	95.5	95.8	88.3	51.2	87.1	89.4	82.9

Overall, these findings suggest that while contemporary models are equipped to handle arithmetic queries, there's significant room for improvement, especially when it comes to intricate mathematical tasks. As research in this domain progresses, refining these models to enhance their accuracy in such tasks will be pivotal.

V. Conclusion

We introduced causal mediation analysis as a mechanism to deeply explore the processing dynamics of Language Models (LMs) particularly in relation to complex arithmetic operations. By orchestrating precise interventions on distinct segments of the model, we evaluated the consequential influence of these segments on the resultant predictions of the model. Our hypothesis centered on the model's strategy for generating responses to intricate arithmetic queries; it transfers mathematically pertinent information from mid-sequence early layers to the concluding token. This information undergoes further processing by the subsequent Multi-Layer Perceptrons (MLP) modules. In validating our theory, we executed experiments grounded in causality on four diverse Transformer-based LMs, yielding empirical data corroborating our proposed model of information flow. Moreover, we illustrated that the identified patterns of information flow were predominantly evident in the context of complex arithmetic inquiries, as opposed to tasks devoid of arithmetic computations. These discoveries not only chart new territories in research pertaining to model refinement, efficient training, and fine-tuning but also emphasize the significance of focusing on the model's specific components responsible for distinct types of inquiries or calculations. In addition, the insights gleaned from this study furnish a foundation for the exploration of corrective measures for LMs on intricate mathematical tasks during inference and allow for the assessment of the credibility of the model's predictions, opening avenues for enhanced reliability in intricate arithmetic problem-solving.

VI. Limitations and Future Work

Our study has provided foundational insights into the inner workings of Language Models (LMs) in handling arithmetic reasoning, with a specific emphasis on complex operations such as exponentiation and roots. However, we acknowledge certain limitations that serve as avenues for further exploration. Primarily, our focus was constrained to equations involving just two operands. It would be insightful for future research to expand this to encompass equations with three or more operands, potentially unveiling more intricate interaction patterns and processing dynamics within the models. Furthermore, while our research complexity was centered on roots and exponentiation, there is a vast landscape of advanced mathematical operations, including derivatives and integrals, which remains to be explored. Such investigations would offer a richer understanding of the models' capabilities in processing diverse and sophisticated mathematical concepts. Additionally, our current approach leaned heavily on interpreting the model's internal mechanisms, and we did not focus on improving its accuracy. As we move forward, integrating interpretative insights with efforts to enhance model accuracy will provide a more holistic understanding of how LMs handle mathematical challenges.

Acknowledgement

The work is supported by the National Science Foundation under NSF Award #2433800, #2100134 and #1946442. This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU. Any opinions, findings, recommendations, expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

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

[2] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[3] T. Liu and B. K. H. Low, "Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks," *arXiv preprint arXiv:2305.14201*, 2023.

[4] D. Spokoyny, I. Lee, Z. Jin, and T. Berg-Kirkpatrick, "Masked measurement prediction: Learning to jointly predict quantities and units from textual context," *arXiv preprint arXiv:2112.08616*, 2021.

[5] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, *et al.*, "Solving quantitative reasoning problems with language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3843–3857, 2022.

[6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought

prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[7] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber, "Investigating gender bias in language models using causal mediation analysis," *Advances in neural information processing systems*, vol. 33, pp. 12388–12401, 2020.

[8] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in gpt," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17359–17372, 2022.

[9] J. Pearl, J. Breese, and D. Koller, "Proceedings of the seventeenth conference on uncertainty in artificial intelligence," 2001.

[10] S. Akter, H. Shahriar, and A. Cuzzocrea, "Towards analysis and interpretation of large language models for arithmetic reasoning," in *IEEE Swiss Conference on Data Science*, 2024. May 30-31.

[11] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward transparent ai: A survey on interpreting the inner structures of deep neural networks," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 464–483, IEEE, 2023.

[12] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," *arXiv preprint arXiv:1506.02078*, 2015.

[13] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[14] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, vol. 3, no. 3, p. e10, 2018.

[15] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter, "Zoom in: An introduction to circuits," *Distill*, vol. 5, no. 3, pp. e00024–001, 2020.

[16] C. Voss, N. Cammarata, G. Goh, M. Petrov, L. Schubert, B. Egan, S. K. Lim, and C. Olah, "Visualizing weights," *Distill*, vol. 6, no. 2, pp. e00024–007, 2021.

[17] R. Langone, A. Cuzzocrea, and N. Skantzos, "Interpretable anomaly prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools," *Data & Knowledge Engineering*, vol. 130, p. 101850, 2020.

[18] K. E. Barkwell, A. Cuzzocrea, C. K. Leung, A. A. Ocran, J. M. Sanderson, J. A. Stewart, and B. H. Wodi, "Big data visualisation and visual analytics for music data mining," in *2018 22nd International Conference Information Visualisation (IV)*, pp. 235–240, IEEE, 2018.

[19] M. Geva, A. Caciularu, K. R. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space," *arXiv preprint arXiv:2203.14680*, 2022.

[20] N. Belrose, Z. Furman, L. Smith, D. Halawi, I. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt, "Eliciting latent predictions from transformers with the tuned lens," *arXiv preprint arXiv:2303.08112*, 2023.

[21] S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders, "Language models can explain neurons in language models," *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2023.

[22] J. Pearl, "Direct and indirect effects," in *Probabilistic and causal inference: the works of Judea Pearl*, pp. 373–392, 2022.

[23] M. Finlayson, A. Mueller, S. Gehrmann, S. Shieber, T. Linzen, and Y. Belinkov, "Causal analysis of syntactic agreement mechanisms in neural language models," *arXiv preprint arXiv:2106.06087*, 2021.

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

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

[26] K. K. Pal and C. Baral, "Investigating numeracy learning ability of a text-to-text transfer model," *arXiv preprint arXiv:2109.04672*, 2021.

[27] P. Piekos, H. Michalewski, and M. Malinowski, "Measuring and improving bert's mathematical abilities by predicting the order of reasoning," *arXiv preprint arXiv:2106.03921*, 2021.

[28] J. Pearl, *Causality*. Cambridge university press, 2009.

[29] D. Martin-Moncunill, M.-A. Sicilia, L. González, and D. Rodríguez, "On contrasting yago with gpt-j: An experiment for person-related attributes," in *Iberoamerican Knowledge Graphs and Semantic Web Conference*, pp. 234–245, Springer, 2022.

[30] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," in *International Conference on Machine Learning*, pp. 2397–2430, PMLR, 2023.

[31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[32] Y. Razeghi, R. L. Logan IV, M. Gardner, and S. Singh, "Impact of pretraining term frequencies on few-shot numerical reasoning," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 840–854, 2022.