Journal of Machine Learning Research 27 (2026) 1-46 Submitted 7/24; Revised 5/26; Published 8/26

Exploring Inference Patterns of LLMs as Symbolic Interactions

Lei Cheng[*] CHENGLEI20020408@SJTU.EDU.CN
School of Computer Science, Shanghai Jiao Tong University[a]

Wen Shen[*] WENSHEN@TONGJI.EDU.CN
School of Computer Science and Technology, Tongji University[b]

Quanshi Zhang[*,1] ZQS1022@SJTU.EDU.CN
School of Computer Science, Shanghai Jiao Tong University[a]

Yuchen Huang HYC18382428684@GMAIL.COM
School of Computer Science, Shanghai Jiao Tong University[a]

Yiming Zhang MXSABER@SJTU.EDU.CN
Department of Electrical and Computer Engineering, Shanghai Jiao Tong University[a]

Yuxiao Yang YANGYUXIAO@SJTU.EDU.CN
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University[a]

Mingjie Li LIMINGJIE0608@SJTU.EDU.CN
Department of Computer Science and Engineering, Shanghai Jiao Tong University[a]

[a] *800 Dongchuan Road, Minhang District, Shanghai 200240, China.*

[b] *4800 Cao'an Highway, Jiading District, Shanghai 201804, China.*

[*] *Equal contributions.*

[1] *Corresponding author.*

Editor: Weijie Su

Abstract

In this paper, we examine whether a set of sparse symbolic interactions can faithfully explain the complex inference patterns of large language models (LLMs), thereby using interactions to diagnose the detailed representations of LLMs. Since previous studies lack a comprehensive proof for interaction-based explanations, we first experimentally verify the sparsity of interactions in LLMs, *i.e.*, we observe that 50–100 interactions are usually powerful enough to accurately mimic all subtle changes in the LLM's output scores over an exponential number of arbitrary masked inputs. More crucially, interactions provide new perspectives for analyzing an LLM. (1) We propose to disentangle the interactions encoded by the LLM in a coarse-to-fine manner and use interactions to analyze the increasing representational complexity of features during forward propagation. The clear disentanglement of interactions enables us to identify incorrect inference patterns hidden in an LLM. (2) We empirically demonstrate that LLMs with different architectures encode similar interactions for text generation, and simple interactions are more likely to be consistently extracted by different LLMs than complex interactions. (3) We also empirically demonstrate that an LLM encodes similar interactions over semantically similar input sentences. Again, simple interactions are more likely to be stably extracted than complex interactions. (4)

©2026 Lei Cheng, Wen Shen, Quanshi Zhang, Yuchen Huang, Yiming Zhang, Yuxiao Yang, and Mingjie Li.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v27/24-1060.html.

We empirically find that low-order interactions can serve as indicators for guiding model compression.

Keywords: explainable artificial intelligence, large language models, interactions, interpretability, deep learning

1. Introduction

This paper is inspired by a new counter-intuitive finding in deep neural networks (DNNs), *i.e.*, it has been found by Ren et al. (2024) that for all DNNs that satisfy three common conditions[1], the DNN typically encodes only a small number of interaction patterns as primitive inference patterns. These interactions represent non-linear relationships among different input variables (*e.g.*, words/tokens in Fig. 1), and each interaction contributes a specific numerical effect to the output score of the DNN.

The notion of interaction is illustrated in Fig. 1. Given an input prompt, "*Einstein's Theory of General Relativity proposes that gravity is the warping of*," the interaction among the words in the set $S = \{\textit{Einstein's}, \textit{warping}, \textit{of}\}$ contributes a numerical effect I_S to boost the DNN's confidence in generating the target word "*space*." Ren et al. (2023a, 2024) have proven a series of theorems, showing that an LLM typically encodes only a small number of interactions per input sample. These interactions can accurately predict the network's outputs across a sufficiently large set of samples, thereby ensuring the faithfulness of the interaction-based explanation.

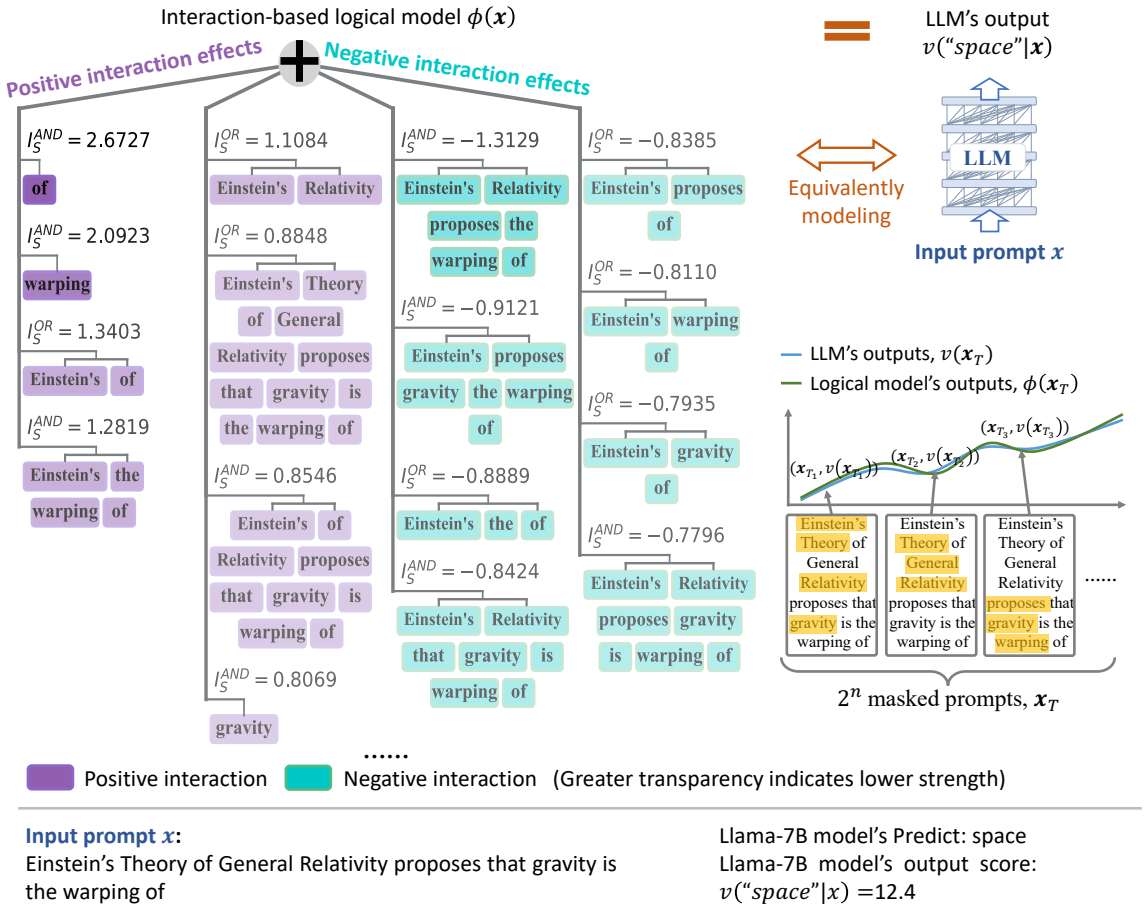
Nevertheless, this study focuses on a more fundamental question: **can interactions faithfully reflect the majority of the inference logic in LLMs?** This remains an open problem.

• LLMs often comprise billions of parameters and are trained on terabytes of data. Therefore, analyzing an LLM's behavior on a given input using only a small number of interactions appears counter-intuitive and lacks sufficient validation.

• Unlike traditional mechanistic interpretability approaches, directly analyzing specific neurons or sub-networks/circuits (Olah et al., 2020; Elhage et al., 2021; Conmy et al., 2023), interactions are encoded by the entire model, rather than being localized to particular neurons. Thus, when the interaction cannot be physically mapped to specific neurons, how to verify the faithfulness or fidelity of such an interaction-based explanation remains an unresolved issue.

• Although previous work has partially proven the sparsity of AND interactions, the three mathematical assumptions underlying it are more like idealized conjectures.[1] These assumptions are also difficult to verify strictly on LLMs.[2]

1. Ren et al. (2024) have proven the sparsity of interactions under three common conditions for *smooth inference on masked/occluded samples*. (Condition 1) The DNN does not encode extremely high-order AND interactions, *e.g.* an interaction between nine words. (Condition 2) The average network output over all samples, $\overline{v}^{(m)} \triangleq \mathbb{E}_{|T|=m}[v(\boldsymbol{x}_T)]$, which have m unmasked input variables, increases when m becomes larger. (Condition 3) When we decrease the number of unmasked input variables, *i.e.*, m, the decreasing speed of the average network output $\overline{v}^{(m)}$ is not faster than a polynomial function of m. Please see Appendix C for detailed introductions of the three conditions.
2. Please see Appendix D for reasons why it is difficult to examine the three conditions[1] in LLMs.



Input prompt $\boldsymbol{x}$:
Einstein's Theory of General Relativity proposes that gravity is the warping of

Llama-7B model's Predict: space
Llama-7B model's output score:
$v(\text{“space”}|\boldsymbol{x})$ =12.4

Figure 1: Illustrating that the LLM function on exponentially many masked samples can be equivalently represented by a logical model based on AND-OR interactions. Given an input sentence $\boldsymbol{x}$ with n words, the interaction patterns in the logic model $\phi(\boldsymbol{x})$ can well predict all subtle changes of the LLM's outputs v on all 2^n randomly masked[3] prompts $\{\boldsymbol{x}_T\}$, *i.e.*, $\forall T \subseteq N, \phi(\boldsymbol{x}_T)$ accurately approximates $v(\boldsymbol{x}_T)$. This guarantees the faithfulness of using interactions to analyze the LLM.

Therefore, in this paper, we aim to validate the faithfulness of interactions in LLMs, and further explore new insights into the complex inference logic of LLMs from the perspective of interactions.

Verifying faithfulness of primitive interactions. We empirically verify the faithfulness of taking interactions as primitive inference patterns, thereby allowing us to identify defective semantic patterns in LLMs. Specifically, we verify that no matter how we randomly mask the input sentence,[3] almost all subtle changes in the LLM's outputs can be accurately mimicked by a logical model based on these interactions, as illustrated in Fig. 1.

3. It is common to use a specific embedding to mask input variables of a DNN (Lundberg and Lee, 2017; Ancona et al., 2019; Fong et al., 2019), but there are still no unified masking strategies. Please see Appendix A for introductions of masking strategies.

More importantly, this approach enables us to go beyond conventional performance metrics and directly assess whether the LLM uses a set of semantically correct interactions for text generation. For example, given the sentence "*Mary is the wife of Adam. James is the son of Adam. Mary is the mother of*," an LLM may incorrectly output "*Jesus*." This hallucination occurs because the model falsely associates the input with religious-related associations in the training data, improperly linking "*Mary*" and "*Adam*" to "*Jesus*." We conduct extensive experiments in Section 5.1 to examine the correctness of detailed inference patterns used by an LLM.

Exploring coarse-to-fine reasoning structure. We propose a method to disentangle interactions for coarse inference logic encoded by an LLM and interactions for fine-grained inference logic encoded by an LLM. We have observed that the LLM typically uses low-order simple interactions (*i.e.*, interactions between a small number of input words) to narrow down the candidate token space to a contextually relevant subset. Subsequently, high-order complex interactions (*i.e.*, interactions between a large number of input words) are usually activated to leverage detailed domain knowledge for final token selection. Example illustrations are provided in Fig. 5.

Exploring how the interactions' complexity changes through forward propagation. We also use interactions to analyze the increasing representational complexities of features during forward propagation. We empirically find that as features propagate forward through successive layers in the LLM, the model progressively constructs increasingly complex interactions.

Analyzing the representation quality of LLMs. Beyond the coarse-to-fine reasoning structure, we propose metrics to evaluate the representational quality of an LLM by analyzing the consistency and stability of interaction patterns encoded by the LLM. The consistency metric measures whether two LLMs with different architectures learn similar interactions. For example, as Fig. 2 shows, given the same sentence "*Einstein's Theory of*^{a} *General Relativity proposes that gravity is the warping of*^{b}," different LLMs usually all encode similar interactions, such as $\{$*Einstein's*, *of*$^a\}$, $\{$*Einstein's*, *Relativity*, *proposes* , *the*, *warping*, *of*$^b\}$, and $\{$*Einstein's*, *the*, *warping*, *of*$^b\}$, to generate the next token "*space*." The stability metric measures whether an LLM learns similar interactions over two semantically similar sentences. We empirically find that low-order interactions exhibit higher consistency and stability than high-order interactions. Furthermore, we also experimentally observe that LLMs with superior generalization power are more likely to mainly encode low-order interactions and avoid learning high-order interactions.

Using interactions to guide model compression. We also explore how interaction patterns can be used to assess the representation quality of a compressed LLM, so as to help determine the optimal pruning rate. We empirically observe a strong correlation between the low-order interactions and model performance. This correlation emphasizes the potential of using low-order interactions as a metric to help determine the optimal pruning ratio.

2. Related Work

Explaining LLMs. Recently, some studies have explained the LLM from different perspectives. Sahu et al. (2022) measured the consistency of an LLM responding to conceptually relevant background knowledge. Some studies observed that many LLMs had fragments

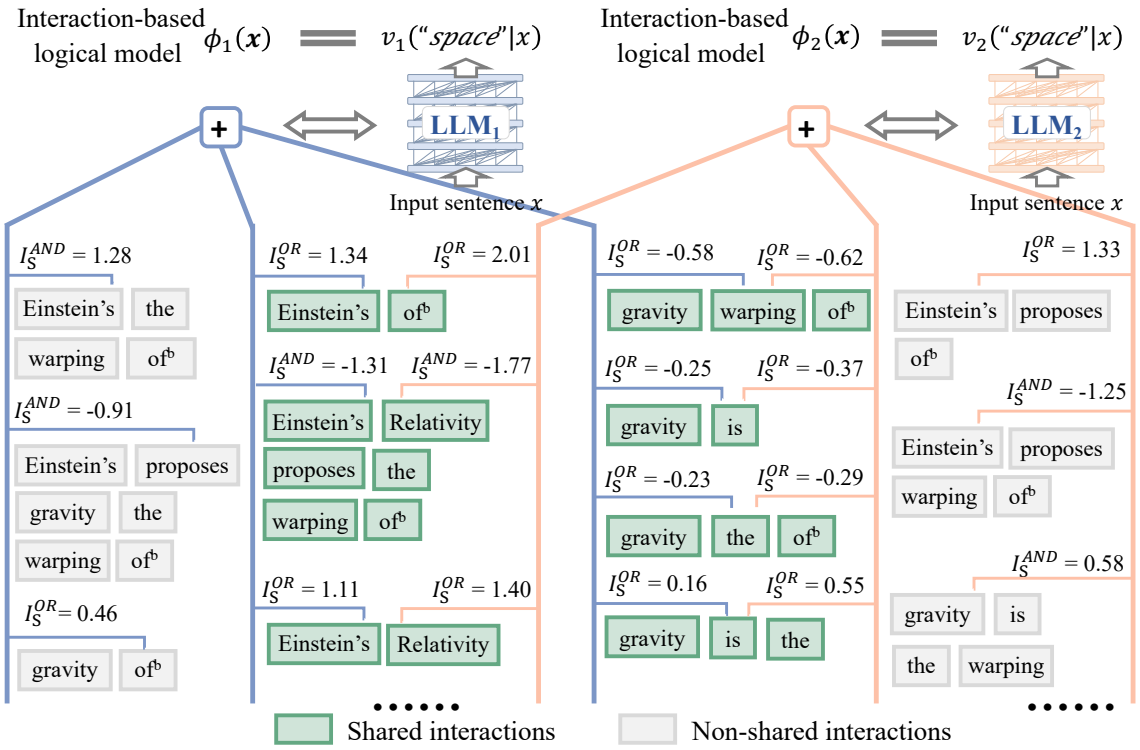


Figure 2: Illustrating that two different LLMs share a large set of salient interactions for inference when given the same input sentence $\boldsymbol{x}$. We show two AND-OR logical models, each containing the top ten most salient interactions encoded by each LLM. We find the seven green interactions (70.0% of all the ten interactions) are shared by two LLMs. The remaining three gray interactions in each logical model are exclusively encoded by a single LLM.

of knowledge (Gu et al., 2022), or observed that features generated by the GPT-3 model were similar to each other (Hansen and Hebart, 2022). Some studies (Dai et al., 2022; von Oswald et al., 2022) related the in-context learning ability of LLMs to running the gradient descent on attention weights. Besides, Bansal et al. (2022) measured gradient-based importance scores *w.r.t.* different layers of an LLM, and observed that different layers contributed unequally to the LLM's ability. Meng et al. (2023) identified neuron activations that had a great impact on the LLM's prediction when the LLM processed input tokens. Wendler et al. (2024) investigated the correlations between features of each intermediate layer and specific natural languages, such as English and Chinese. However, previous studies failed to clarify the exact mathematical mechanism behind the explanation. In comparison, in this paper, we analyze the output score of an LLM using a small number of interactions, which has been proved to be a faithful explanation.

Using game-theoretic interactions to analyze DNNs. Traditional explanations, relying on visualization (Tenney et al., 2020) or interpretable surrogate models (Zhang et al., 2020), did not formulate and mathematically guarantee that the explanation result really reflected the true logic used by the DNN with a theoretical error bound. This might be why

Rudin (2019) calls for us to stop explaining black-box models for high stakes decisions, and shift attention towards constructing inherently interpretable models.

To this end, using interactions between input variables of a DNN presents a new direction to analyze a DNN, which has a series of solid theories to ensure the correctness of the explanation (Ren et al., 2023a). Some studies used interactions to **investigate the representation power of a DNN**, such as analyzing the adversarial robustness and adversarial transferability (Wang et al., 2021a,b), investigating the overfitting problem (Zhou et al., 2023b; Ren et al., 2023c), and proving the bottleneck of representing specific interactions (Deng et al., 2022). The interaction was also used to **analyze the common mechanism shared by many empirical deep learning methods.** Deng et al. (2024) proved that the core mechanisms of fourteen attribution methods could be reformulated as a reallocation of interactions. Zhang et al. (2022a) proved the common mechanism shared by twelve previous transferability-boosting methods, that is, these methods all reduced interactions between regional adversarial perturbations. Ren et al. (2023a); Li and Zhang (2023a) discovered and Ren et al. (2024) proved that a DNN usually encoded sparse interactions. However, the proved conditions[1] for the sparsity of interactions are difficult to examine on the text generation task (see Appendix D for reasons). Thus, we still need to verify the faithfulness of using interactions to analyze LLMs in this paper.

Interaction-based explanation vs. mechanistic interpretability. Mechanistic interpretability (Olah et al., 2020; Elhage et al., 2021; Conmy et al., 2023) aims to explain DNNs by deconstructing them into interpretable components—ranging from individual neurons and attention heads to emergent concept circuits—to characterize their functional roles and semantic alignment in the model's overall behavior. In contrast, the approach of interaction-based explanation does not seek to atomize the DNN. Instead, it treats the DNN as an integrated whole and describes its inference logic through a set of AND-OR interactions, such that a DNN's inference logic can be considered equivalent to these AND-OR interaction patterns.

3. Extracting Interactions Used by LLMs

In this section, we conduct experiments to verify that a small set of sparse AND-OR interactions can fully mimic the outputs of the LLM under an exponential number of occluded input states, thereby validating the faithfulness of using interactions to explore new insights into the complex inference logic of LLMs, including the coarse-to-fine inference process and the layer-wise inference logic.

3.1 Preliminaries: Extracting Interactions Used by a DNN

Given a trained DNN, let $\boldsymbol{x} = [x_1, x_2, ..., x_n]^\top$ denote the input sample with n variables, which are indexed by $N = \{1, 2, ..., n\}$. Let $v(\boldsymbol{x}) \in \mathbb{R}$ denote a *scalar* output of the DNN. Take an LLM for text generation for example. Given an input sentence $\boldsymbol{x}$ = "*he is a green hand means that he is a*" with n words, the LLM generates the next token y^{truth}, *e.g.*, $y^{\text{truth}} = [\textit{beginner}]$, by calculating the probability of generating y^{truth}, $p(y = y^{\text{truth}}|\boldsymbol{x})$. Then, the *scalar* output of the LLM can be formulated as the confidence score of generating y^{truth}

as follows.

$$v(\boldsymbol{x}) = \log \frac{p(y = y^{\text{truth}}|\boldsymbol{x})}{1 - p(y = y^{\text{truth}}|\boldsymbol{x})}. \tag{1}$$

Problem setting. In recent years, the interaction has been taken as a new mathematical tool that effectively disentangles primitive inference patterns used by a DNN, and has received increasing attention (Wang et al., 2021a,b; Zhou et al., 2023b; Ren et al., 2023c; Deng et al., 2022; Ren et al., 2023a; Deng et al., 2024). Interaction-based explanation methods employ a logical model $\phi(\cdot)$ as a surrogate to represent the DNN's primitive inference patterns. The faithfulness of the logical model's explanation of the DNN's outputs is guaranteed by two criteria: **(1) Fidelity criterion**: The logical model $\phi(\cdot)$ must accurately capture the DNN's outputs $v(\cdot)$ over a sufficiently large and diverse set of samples Λ; **(2) Conciseness criterion:** The logical model should comprise only a small number of inference patterns to ensure interpretability. These two criteria can be formally defined as follows.

$$\forall \boldsymbol{x}' \in \Lambda, \quad \phi(\boldsymbol{x}') = v(\boldsymbol{x}'), \qquad \text{subject to} \;\; \text{complexity}(\phi) \leq M, \tag{2}$$

where M denotes the maximum allowed complexity for the logical model $\phi(\cdot)$.

First, the logical model $\phi(\cdot)$ is constructed to encode the AND-OR interactions between input variables in N.

$$\forall \boldsymbol{x}' \in \Lambda, \quad \phi(\boldsymbol{x}') \overset{\text{def}}{=} \sum\nolimits_{S\subseteq N} I_S^{\text{AND}} \cdot \mathbf{1}_{\text{AND}}(S|\boldsymbol{x}') + \sum\nolimits_{S\subseteq N} I_S^{\text{OR}} \cdot \mathbf{1}_{\text{OR}}(S|\boldsymbol{x}') + b. \tag{3}$$

The AND trigger function $\mathbf{1}_{AND}(S|\boldsymbol{x}') \in \{0, 1\}$ represents an AND relationship between input variables in S, which is also termed an *AND interaction pattern*. Please see Fig. 1 for examples of AND-OR interactions. If all variables in S are present (not masked) in $\boldsymbol{x}'$, then the AND interaction S is triggered, and $\mathbf{1}_{\text{AND}}(S|\boldsymbol{x}') = 1$; otherwise, the AND interaction S is not triggered, and $\mathbf{1}_{\text{AND}}(S|\boldsymbol{x}') = 0$. I_S^{AND} is the scalar weight of the AND interaction. *The OR trigger function* $\mathbf{1}_{OR}(S|\boldsymbol{x}') \in \{0, 1\}$ represents an OR relationship between input variables in S, which is also termed an *OR interaction pattern*. If any variable in S is present (not masked) in $\boldsymbol{x}'$, then the OR interaction S is triggered, and $\mathbf{1}_{\text{OR}}(S|\boldsymbol{x}') = 1$; otherwise, the OR interaction S is not triggered, and $\mathbf{1}_{\text{OR}}(S|\boldsymbol{x}') = 0$. I_S^{OR} is the scalar weight of the OR interaction. **A video demo is attached as a supplementary material to introduce the logical model.**

Two examples to illustrate the AND trigger function and OR trigger function. Given a sentence $\boldsymbol{x}$ = "he is a green hand," the co-occurrence of the words in $S_1 = \{\text{green}, \text{hand}\}$ yields a numerical interaction contribution $I_{S_1}^{\text{AND}}$ that pushes the DNN's inference toward the meaning of a "beginner." Conversely, if any word in S_1 is masked to produce $\boldsymbol{x}_{\text{masked}}$, the AND trigger function outputs 0, *i.e.* $\mathbf{1}_{\text{AND}}(S_1|\boldsymbol{x}_{\text{masked}}) = 0$, indicating that the network output does not receive this contribution. Similarly, consider sentiment classification of the sentence "it has a poor environment and bad taste," the presence of **either** word in $S_2 = \{\text{poor}, \text{bad}\}$ contributes an interaction effect $I_{S_2}^{\text{OR}}$ toward the negative sentiment. Conversely, the effect is nullified only if both words are masked to produce $\boldsymbol{x}_{\text{masked}}$, in which case the OR trigger function outputs 0, *i.e.* $\mathbf{1}_{\text{OR}}(S_2|\boldsymbol{x}_{\text{masked}}) = 0$, indicating that the network output does not receive this contribution.

Second, the fidelity criterion is guaranteed by the universal matching property of interactions in Theorem 1. Under a specific configuration of the interaction weights I_S^{AND} and I_S^{OR}, the logical model $\phi(\cdot)$ can precisely reproduce the DNN's outputs

$v(\cdot)$, no matter how we randomly mask[3] the input sample $\boldsymbol{x}$. This precise reproduction holds for **all samples within the set $\Lambda = \{\boldsymbol{x}_T | T \subseteq N\}$, which comprises 2^n masked samples and is thus sufficiently large.** Here, $\boldsymbol{x}_T$ denotes a masked variant of the sample $\boldsymbol{x}$, in which all input variables in T remain unchanged, while those in $N \setminus T$ are masked.[4] Theorem 1 theoretically guarantees the faithfulness of considering interactions as primitive inference patterns *equivalently* utilized by the DNN.

Theorem 1 *(Universal matching property; see (Chen et al., 2024; Zhou et al., 2023a)). When scalar weights in the logical model $\phi(\cdot)$ are set to $\forall S \subseteq N$, $S \neq \emptyset$, $I_S^{AND} = \sum_{T \subseteq S} (-1)^{|S|-|T|} \cdot u_T^{AND}$, $I_S^{OR} = -\sum_{T \subseteq S} (-1)^{|S|-|T|} \cdot u_{N \setminus T}^{OR}$, subject to $u_T^{AND} = 0.5v(\boldsymbol{x}_T) + \gamma_T$, $u_T^{OR} = 0.5v(\boldsymbol{x}_T) - \gamma_T$, $b = v(\boldsymbol{x}_\emptyset)$, where $\{\gamma_T\}$ are learnable parameters. Then, no matter how we set $\{\gamma_T\}$, we have*

$$\forall \boldsymbol{x}' \in \Lambda, \quad \phi(\boldsymbol{x}') = v(\boldsymbol{x}'). \tag{4}$$

Theorem 1 provides a group of logical models with respect to different settings of $\{\gamma_T\}$ that all satisfy the universal matching property. Thus, we can follow Li and Zhang (2023b) to optimize parameters $\{\gamma_T\}$ using a LASSO-like loss to pursue the sparsest interactions $\min_{\{\gamma_T\}} \sum_S |I_S^{\text{AND}}| + |I_S^{\text{OR}}|$.

Third, the conciseness criterion is ensured by the sparsity property of interactions. The sparsity of interactions means that only a few interaction terms have salient interaction effects I_S^{AND} or I_S^{OR}, and all other interactions have ignorable effects on the logical model's output, *i.e.*, $I_S^{\text{AND}} \approx 0$, $I_S^{\text{OR}} \approx 0$. This phenomenon has been widely observed in various DNNs trained for different tasks (Ren et al., 2023a; Li and Zhang, 2023a). The sparsity property of AND interactions has been proven by Ren et al. (2024) when the DNN satisfies three common conditions[1].

AND-OR interactions vs. the attention mechanism of Transformers. The attention layer in the Transformer model only analyzes the local signal passing through an intermediate layer. In comparison, the AND-OR interaction represents the overall non-linear relationship encoded by all attention layers in the entire DNN.

Order (complexity) of an interaction. The order k of an interaction S is defined as the number of variables in S, *i.e.*, $k = |S|$. The order of an interaction reflects the complexity of the interaction. The low-order interaction indicates the simple interaction between a small number of input variables, and the high-order interaction indicates the complex interaction between many input variables.

3.2 Empirically Examining the Faithfulness of Interactions in LLMs

Whether the complex inference logic of LLMs can be interpreted in a symbolic manner remains an open problem, particularly since its representation via symbolic interaction patterns seems inherently counterintuitive.

• LLMs with massive parameters trained on vast amounts of data are generally believed to encode a vast number of knowledge points.

• Theoretically, it is difficult[2] to directly verify whether an LLM satisfies the three sufficient conditions[1], further complicating the justification for LLMs' symbolic explanations.

4. In this paper, we generate the masked sentence $\boldsymbol{x}_T$ by using a padding token to mask all tokens belonging to words in $N \setminus T$. The padding token is manually assigned or learned (see Appendix F.1 for details).

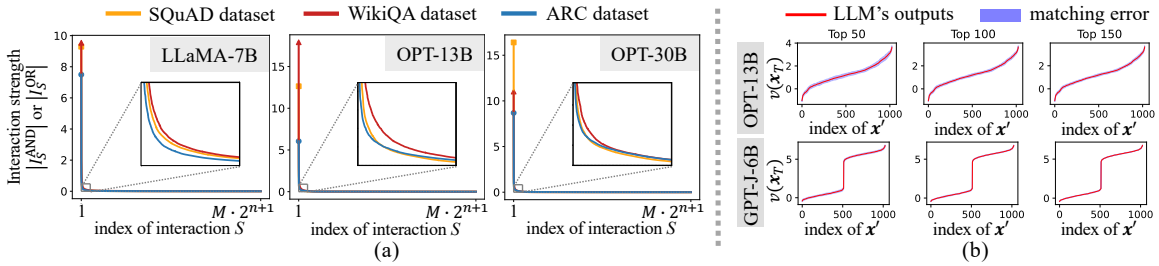


Figure 3: (a) Sparsity of AND-OR interactions extracted from different input samples. We show absolute values of AND-OR interaction effects in a descending order. (b) Verifying the universal matching property. The red line shows model outputs on all 2^n masked states of a single input sentence in an ascending order. We varied the threshold τ to obtain exactly 50, 100, and 150 top-ranked salient interactions, respectively. It shows that when we use more than 100 top-ranked interactions, the approximated outputs match well with the real outputs on different masked sentences. Appendix F.7 shows more results.

• Additionally, the proof provided by Ren et al. (2024) is limited to AND interactions and does not account for OR interactions, raising a further complication.

Therefore, **in this study, we empirically examine the faithfulness of interactions in an LLM**. The following two experiments are designed to provide empirical evidence on this matter. **More crucially, we observe that, contrary to intuition, LLMs with more parameters tend to encode fewer salient interactions.**

Experiment 1. To verify the faithfulness of interactions, we use the following infidelity metric for the explanation based on a small logical model, which is constructed using only K top-ranked salient interactions.

$$\epsilon_{inf}(\boldsymbol{x}') = |v(\boldsymbol{x}') - \hat{\phi}(\boldsymbol{x}')|,$$
$$\forall \boldsymbol{x}' \in \Lambda, \quad \hat{\phi}(\boldsymbol{x}') = \sum\nolimits_{S\in\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}')} I^{\text{AND}}_S \cdot \mathbf{1}_{\text{AND}}(S|\boldsymbol{x}') + \sum\nolimits_{S\in\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}')} I^{\text{OR}}_S \cdot \mathbf{1}_{\text{OR}}(S|\boldsymbol{x}') + bias$$
$$\text{subject to} \quad \Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}') = \{S : |I^{\text{AND}}_S| \geq \tau\}, \quad \Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}') = \{S : |I^{\text{OR}}_S| \geq \tau\}, \tag{5}$$

where τ is set to the K^{th} largest value among $|I^{\text{AND}}_S|, |I^{\text{OR}}_S|$. We construct the logical model $\hat{\phi}$ by using salient interaction sets $\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}')$ and $\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}')$ of varying sizes. This is achieved by setting the thresholds τ to the K^{th} largest value among the interaction effects $|I^{\text{AND}}_S|$ and $|I^{\text{OR}}_S|$, thereby controlling the number of selected interactions. We set $bias = \arg\min_{bias} \mathbb{E}_{\boldsymbol{x}'\in\Lambda}[|\hat{\phi}(\boldsymbol{x}') - v(\boldsymbol{x}')|^2]$.

In this way, Fig. 3 illustrates the infidelity $\epsilon_{inf}(\boldsymbol{x}')$ (matching error $|v(\boldsymbol{x}') - \hat{\phi}(\boldsymbol{x}')|$) in approximating the LLM outputs, when we construct logical models of different sizes for evaluation. It shows that when we use more than 100 top-ranked interactions, the approximated outputs match well with the real outputs on different masked sentences. The averaged infidelity over different LLMs on different sentences is quite small. This proves that the output score of the LLM can be represented with sparse interactions.

Experiment 2: *Fig. 3(a) visualizes the distribution of interaction effects encoded by different LLMs to examine the sparsity of interactions.* Given each input sentence $\boldsymbol{x}$, we

obtain 2^n AND interactions and 2^n OR interactions. In this way, given M testing sentences, we can collect a total of $M \cdot 2^{n+1}$ interactions.

Each subfigure in Fig. 3(a) sorts absolute values of all these AND-OR interactions in descending order. Only a few AND-OR interactions among all AND-OR interactions have significant effects on the LLM's outputs, and all other AND-OR interactions have almost zero effect on the LLM's outputs. Therefore, it is verified that an LLM's output scores can be disentangled into a small number of interactions.

Experimental details. We use interactions to analyze six LLMs for text generation, including the OPT-1.3B/13B/30B model (Zhang et al., 2022b), the GPT-J-6B model (Wang and Komatsuzaki, 2021), the GPT-Neo-2.7B model (Black et al., 2021), and the LLaMA-7B model (Touvron et al., 2023a). Appendix F.1 shows details about how to mask words for these LLMs. For some SOTA LLMs, *e.g.*, ChatGPT, we cannot obtain the entire output probability vector, so we cannot compute AND-OR interactions on these LLMs. We analyze all LLMs on three data sets,[5] including the WikiQA (Yang et al., 2015), SQuAD (Rajpurkar et al., 2016), and ARC (Clark et al., 2018) data sets. We take the first few words of each document in each data set as the input of an LLM, and we let the LLM complete the input.[6] We follow Chen et al. (2024) to select $n = 10$ meaningful words from each input sentence to compute interactions, so as to reduce the computational cost.[7] The remaining unselected words are never masked, just like a constant background.

3.3 Technology Extension 1: Coarse-to-Fine Interaction Analysis in LLMs

In general, the inference logic of an LLM is highly complex. Therefore, in this subsection, we extend interaction-based explanations to decompose the complex inference logic of LLMs into two categories: coarse-grained logic and fine-grained logic.

• Coarse-grained interactions refer to the interactions used by the LLM to generate a small domain of contextually related tokens. These interactions usually use co-occurrence relationships between tokens to identify a set of related tokens, just like the local strategy of the bag-of-words model. For example, given a sentence $\boldsymbol{x}$ = "*The theory of relativity is centered around the speed of*," the interactions used by the LLM to generate tokens such as "*matter*," "*planet*," "*particle*," and "*light*" are considered coarse-grained interactions.

We set $v^{\text{coarse}}(\boldsymbol{x}) = \log \frac{p(Y^{\text{related}}|Y^{\text{all}})}{1-p(Y^{\text{related}}|Y^{\text{all}})}$ to denote the LLM's output that mainly relies on **coarse interaction patterns**, where $p(Y^{\text{related}}|Y^{\text{all}}) = \sum_{y\in Y^{\text{related}}} p(y|Y^{\text{all}})$. Y^{all} denotes the set of all tokens encoded by the LLM. $Y^{\text{related}} \subset Y^{\text{all}}$ denotes a subset of tokens that are relevant to the domain knowledge (*e.g.*, tokens related to physics: *matter*, *planet*, *particle*,

5. Note that regardless of whether the documents in these data sets were originally collected for text generation, our experiments repurposed them for text generation tasks. Specifically, we use the initial portion of each document as input to the LLM, which then performs text generation. Please see Appendix F.4 for details about experimental data.
6. We take words, instead of tokens, as input variables in N, because different LLMs may divide the same word into different tokens. For example, the LLaMA-7B model divides the word "*raining*" into two tokens "*ra*" and "*ining*," while the OPT-1.3B model obtains a single token "*raining*." Thus, we take all tokens within a word as a single input variable for fair comparisons.
7. Here, meaningful words refer to those words that are **not** conjunctions, prepositions, articles, and other function words, such as "is," "a," "the," and so on, because they are usually not so informative for language generation as other words. More importantly, as shown by Chen et al. (2024), the selection of meaningful words will not fundamentally compromise the faithfulness of interactions.

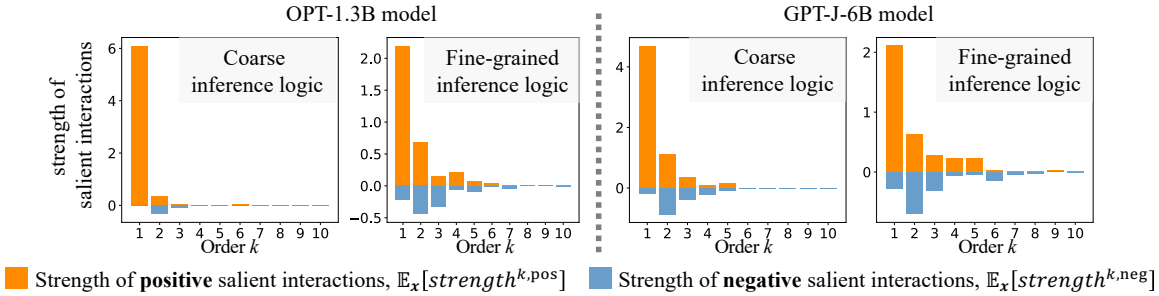


Figure 4: Comparing the distribution of salient interactions over different orders (complexities) when the LLM conducts coarse-grained inference and that when the LLM conducts fine-grained inference. Results show that when the LLM conducts fine-grained inference, the LLM usually utilizes more high-order (complex) interactions compared to the case when the LLM conducts coarse inference.

and *light* in the above example).[8] $p(y|Y^{\text{all}})$ is the output probability of the LLM generating the token y.

• Fine-grained interactions refer to the interactions employed by the LLM to generate the final token "*light*." Therefore, in this subsection, we design a new interaction metric to disentangle two distinct types of interactions.

We set $v^{\text{fine}}(\boldsymbol{x}) = \log \frac{p(y^{\text{truth}}|Y^{\text{related}})}{1-p(y^{\text{truth}}|Y^{\text{related}})}$ to denote the LLM's classification confidence of using **fine-grained interaction patterns** to identify the exact target token y^{truth} from relevant tokens in Y^{related}. The probability $p(y^{\text{truth}}|Y^{\text{related}})$ is computed by putting output scores of tokens in Y^{related} to a softmax layer.

We use the following metrics to represent the strength of interactions of the k-th order.

$$strength^{k,\text{pos}} = \sum_{S\in\Omega^{\text{AND}}_{\text{salient}},|S|=k} \max(I^{\text{AND}}_S, 0) + \sum_{S\in\Omega^{\text{OR}}_{\text{salient}},|S|=k} \max(I^{\text{OR}}_S, 0), \tag{6}$$

$$strength^{k,\text{neg}} = \sum_{S\in\Omega^{\text{AND}}_{\text{salient}},|S|=k} \min(I^{\text{AND}}_S, 0) + \sum_{S\in\Omega^{\text{OR}}_{\text{salient}},|S|=k} \min(I^{\text{OR}}_S, 0). \tag{7}$$

Fig. 4 shows the distribution of interactions over different orders. We have observed that **compared to the coarse inference, the fine-grained inference encodes more complex (higher-order) interactions to represent fine-grained domain knowledge**, *e.g.*, the knowledge of "*the theory of relativity.*"

Fig. 5 shows how the OPT-1.3B model makes inference in a coarse-to-fine manner. Results show that among salient interactions, the coarse interactions are mainly of low orders (from the first order to the third order). In comparison, the fine-grained interactions include not only lower-order interactions but also higher-order interactions, such as ninth-order and tenth-order interactions.

8. Please see Appendix F.5 for details about how to select domain-specific tokens and salient interactions.

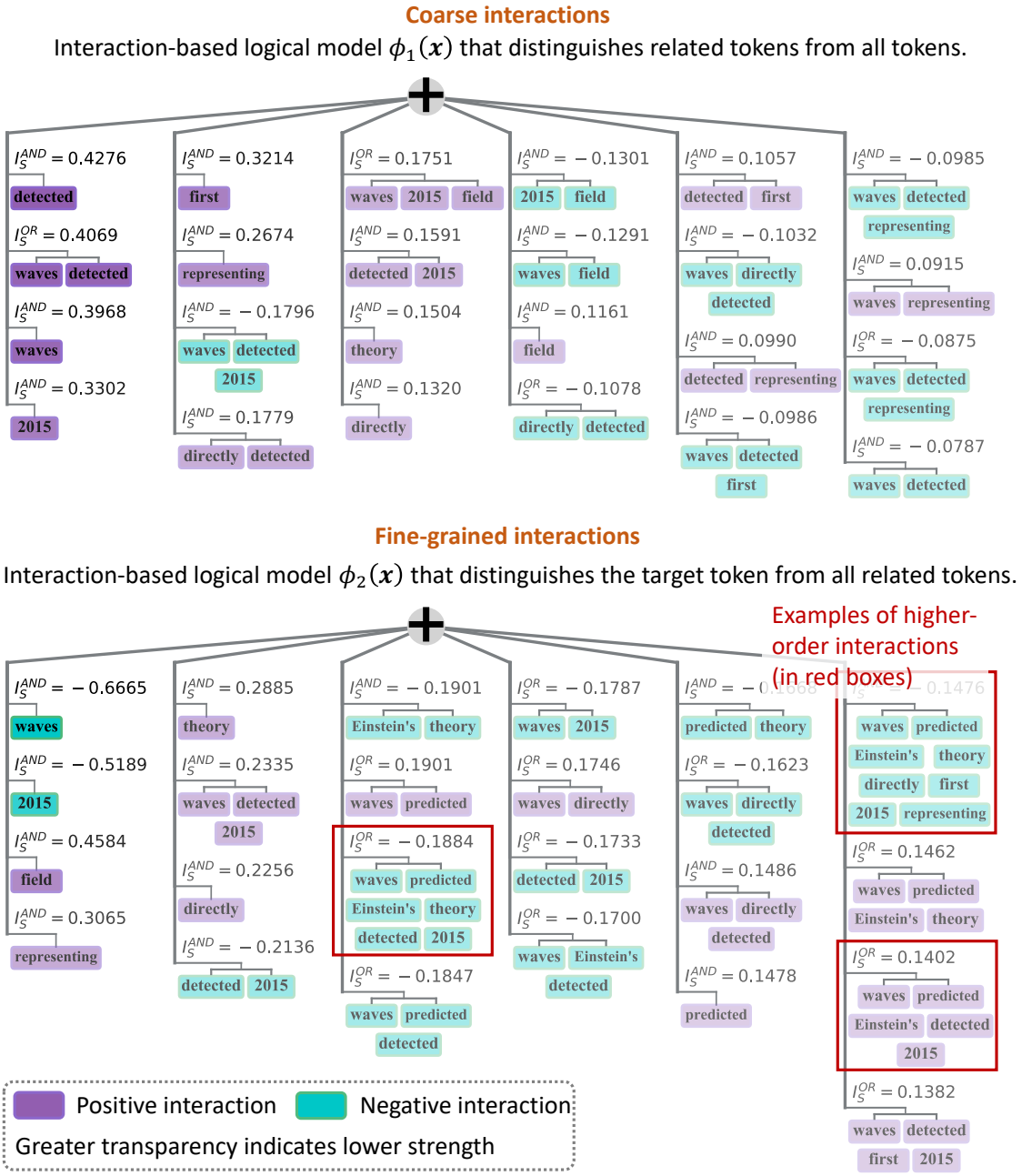


Figure 5: Comparing interactions used for coarse-grained inference with those used for fine-grained inference. The color intensity represents the interaction strength. Results indicate that coarse-grained inference primarily uses low-order interactions, while fine-grained inference incorporates both low-order and high-order interactions.

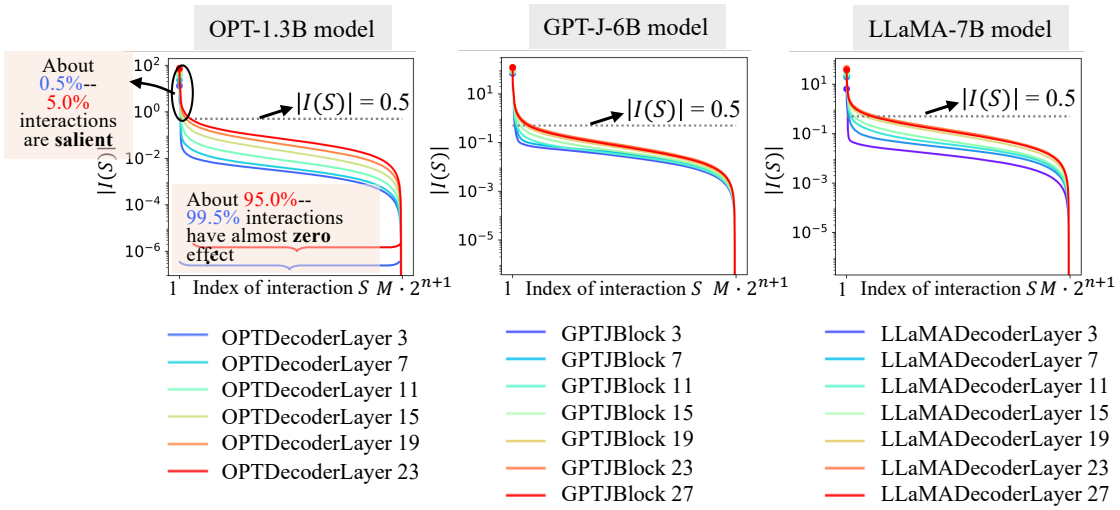


Figure 6: Gradually encoding new AND-OR interactions during forward propagation of an LLM.[10] The blue curve represents interactions encoded by features in low layers, and the red curve represents interactions encoded by features in high layers. The strength $|I_S^{\text{AND}}|$ or $|I_S^{\text{OR}}|$ of all interactions extracted from M input samples is shown in the **log** space in a descending order.

3.4 Technology Extension 2: Layer-Wise Interaction Analysis in LLMs

Traditional interaction-based explanations typically focus on interpreting the interaction patterns utilized by the entire DNN. In this subsection, we extend interaction-based explanations to analyze the evolution of an LLM's complex inference logic through its layer-wise forward propagation.

Specifically, we define the explanatory target as identifying the interactions "used" in generating features at different layers. This research shows how an LLM uses low-order (simple) interactions to compute features in low layers, and gradually uses interactions of higher orders (more complex ones) to compute features in high layers. Specifically, for each input sentence $\boldsymbol{x}$, the explanation target for the l-th layer, *i.e.,* $v^l(\boldsymbol{x})$, is defined as $v^l(\boldsymbol{x}) = o^l(\boldsymbol{x})^\top f^l(\boldsymbol{x}')$, where $f^{(l)}(\boldsymbol{x}')$ denotes the feature vector of the l-th layer. $o^l(\boldsymbol{x}) = f^l(\boldsymbol{x})/\|f^l(\boldsymbol{x})\|_{\text{L2-norm}}$ represents the feature direction of the original input. In this way, we can consider the interactions computed on $v^l(\boldsymbol{x})$ as the inference patterns utilized by the LLM to generate the l-th layer's features.

We conduct experiments to extract and compare interactions encoded by different layers of each LLM on the SQuAD data set. We follow the settings of Experiment 2 in Section 3.2 to visualize the strength of interactions in a descending order. Fig. 6 compares the sparsity of interactions extracted from different layers. It shows that computing features of higher layers generally requires more interactions, *i.e.*, **the LLM continuously learns new salient interactions through the forward propagation process**. More crucially, Fig. 7 shows that features in low layers tend to represent low-order (simple) interactions. In comparison, features in high layers usually model both low-order interactions and higher-

10. Please see Appendix F.3 for details about network architectures and chosen layers.

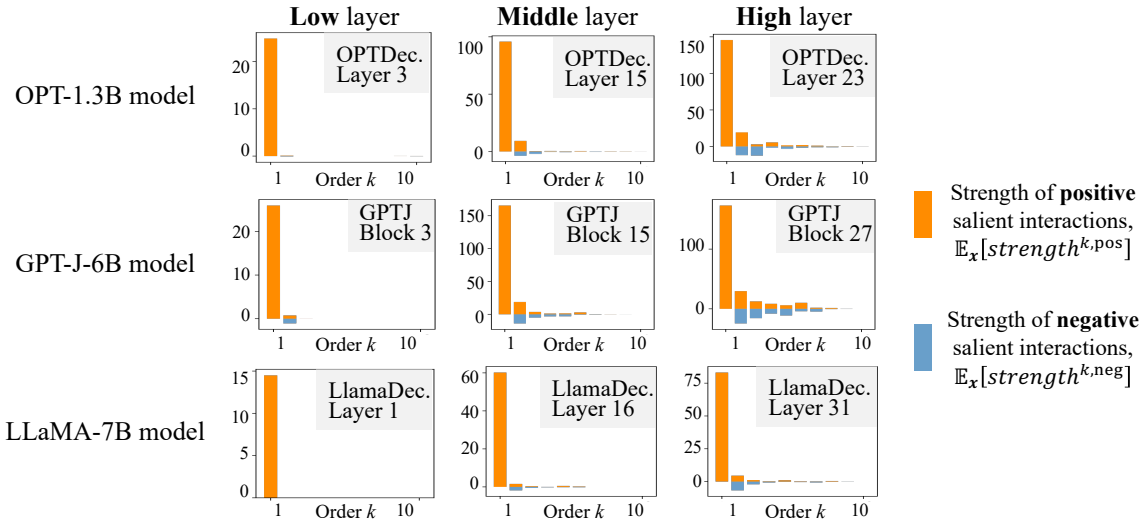


Figure 7: Comparing orders (complexities) of AND-OR interactions extracted from different layers of an LLM. Results show that high-order interactions gradually emerge through the forward propagation process.

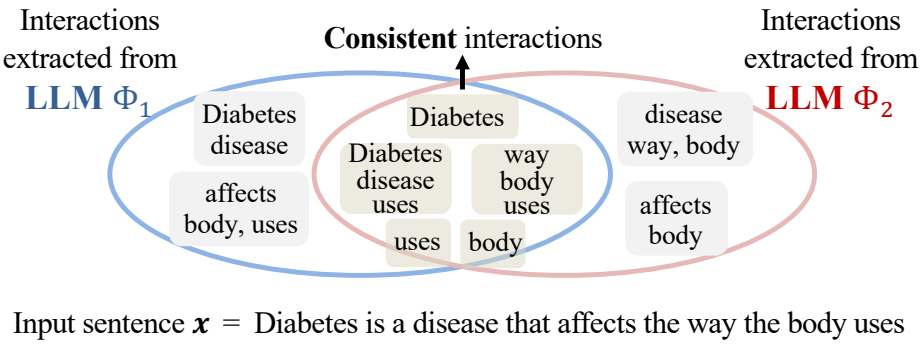


Figure 8: Illustration of consistent interactions encoded by different LLMs.

order interactions. This reflects **the gradual emergence of high-order interactions in the forward propagation process**.

4. Interaction Metrics for the Representation Quality of LLMs

In this section, we aim to propose metrics to evaluate the representational quality of an LLM by analyzing the consistency and stability of interaction patterns encoded by the LLM. The consistency metric measures whether two LLMs with different architectures learn to converge to similar sets of salient interactions. The stability metric measures whether a salient interaction is a stable inference pattern over semantically similar sentences.

4.1 Consistency of Interactions over LLMs

Quantifying the consistency of interactions is crucial for analyzing an LLM. This is because if an interaction pattern is universally shared among all LLMs, then this interaction pattern tends to be highly convincing, and we refer to such patterns as *consistent interaction patterns*. Conversely, inconsistent interaction patterns represent abnormal patterns that are encoded only in a single LLM or in very few LLMs. Inconsistent interaction patterns often reveal systemic biases or representational defects, which may stem from variations in training data, model architecture, or optimization objectives. In this journal paper, we were inspired by and extended our previous work (Lou et al., 2026). Specifically, we extracted interactions from six LLMs for text generation tasks using the WikiQA, SQuAD, and ARC data sets (details of these models and data sets can be found in Section 3.2) and another six LLMs for text generation on the BBC News data set (Greene and Cunningham, 2006). The six models include: Qwen2.5-7B/14B (Qwen et al., 2024), Llama-3.1-8B (Grattafiori et al., 2024), DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI, 2025), DeepSeek-R1-Distill-Qwen-7B, and DeepSeek-R1-Distill-Qwen-14B.

As Fig. 8 shows, given the same input sentence $\boldsymbol{x}$, if two different LLMs Φ_1 and Φ_2 extract similar AND interactions and similar OR interactions, *i.e.*, $\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_1) \approx \Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_2)$, $\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_1) \approx \Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_2)$, then we can consider that the learning of these two LLMs converges to consistent representations. Those shared interactions are considered to be consistent representations encoded across different LLMs.

Therefore, we propose the following metric to quantify the consistency of interactions encoded by the LLM Φ_1 as the ratio of salient interactions encoded by Φ_1 that are also encoded by Φ_2.

$$sim(\Phi_1;\Phi_2|\boldsymbol{x};\alpha_1;\alpha_2)=\frac{|\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)\cap\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_2;\alpha_2)|+|\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)\cap\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_2;\alpha_2)|}{|\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)|+|\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)|}, \tag{8}$$

where the two sets of salient interactions encoded by the LLM Φ_1 are selected by following Li and Zhang (2023a), as follows.

$$\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_1,\alpha_1)=\left\{S\subseteq N:|I^{\text{AND}}_S|>\tau(\alpha_1)\right\}, \tag{9}$$

$$\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_1,\alpha_1)=\left\{S\subseteq N:|I^{\text{OR}}_S|>\tau(\alpha_1)\right\}, \tag{10}$$

$$\text{s.t.}\quad \tau(\alpha_1)\overset{\text{def}}{=}\alpha_1\cdot\max\{|I^{\text{AND}}_{S_1}|,...,|I^{\text{AND}}_{S_n}|,|I^{\text{OR}}_{S_1}|,...,|I^{\text{OR}}_{S_n}|\}. \tag{11}$$

We try different thresholds $\tau(\alpha_1)$ by setting different values of $\alpha_1\in[0.05,0.3]$ to collect interactions of different saliency. Similarly, for the LLM Φ_2, we set $\alpha_2=0.05$ to compute the threshold for selecting salient interactions $\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}|\Phi_2,\alpha_2)$ and $\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}|\Phi_2,\alpha_2)$.

In experiments, we evaluate the interaction consistency by using a set of testing prompts selected[11] from those that make the two LLMs Φ_1 and Φ_2 generate the same token in the experiments described in Section 3.2. Fig. 9 shows that the consistency of interactions over different LLMs increases along with the increase of the saliency parameter α_1. This implies that the more salient an interaction is, the more likely it is to remain consistent over different LLMs. When we set $\alpha_1=0.2$, on average, 71.72% of salient interactions are consistent across different LLMs. This proves the relatively high consistency of salient interactions over different LLMs.

Fig. 10 visualizes consistent and inconsistent interactions. We can see that consistent interactions tend to be low-order (simple) interactions, while higher-order (complex)

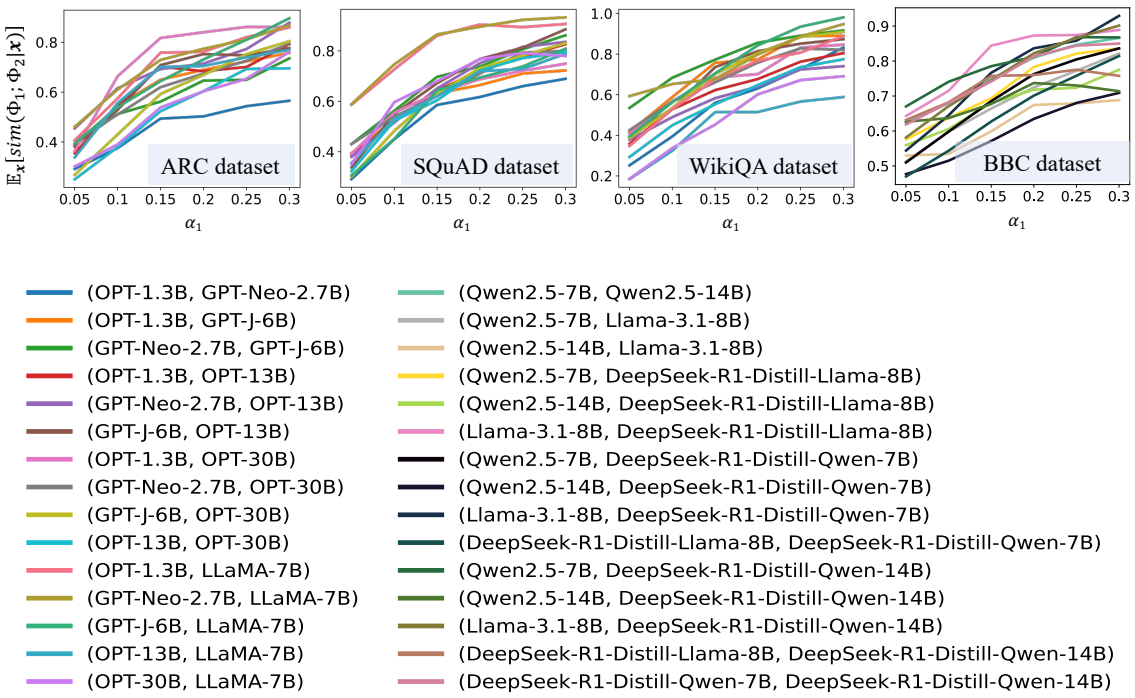
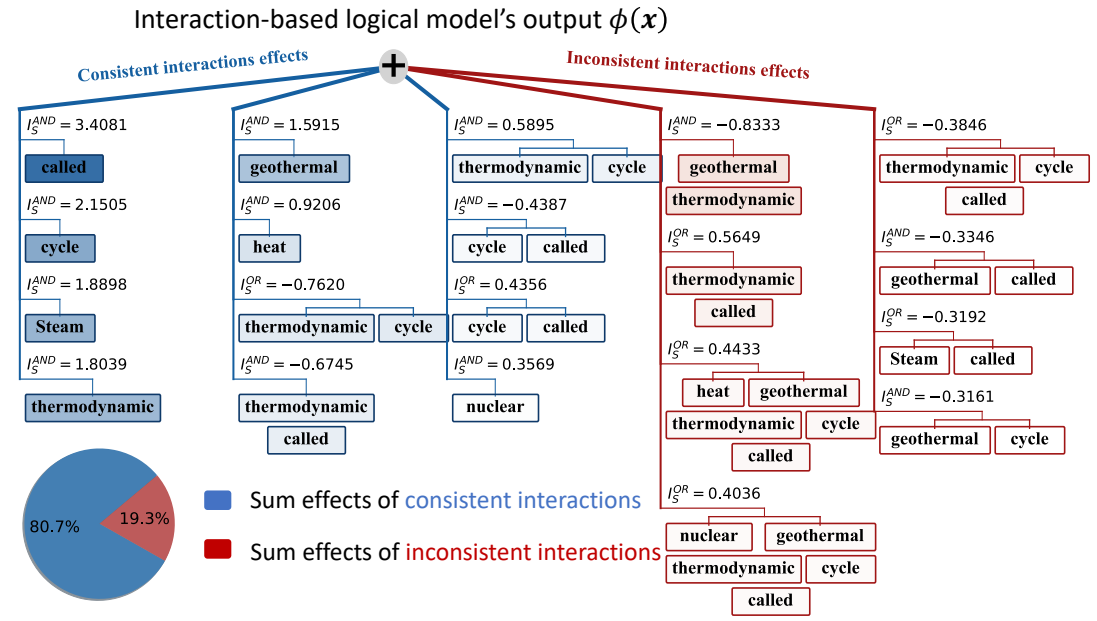


Figure 9: Consistency of interactions $sim(\Phi_1;\Phi_2|\boldsymbol{x};\alpha_1;\alpha_2)$ for each pair of LLMs. For each pair of LLMs, we compute consistency values across different saliency parameters α_1. We report the average consistency value over different input prompts. Results show that interactions of higher saliency usually have higher consistency values over LLMs.

interactions frequently exhibit inconsistency. This observation motivates us to further investigate the consistency of interactions of each specific order. To this end, we use the following metric to measure the consistency of interactions of the k-th order, *i.e.*, $sim^{(k)}(\Phi_1;\Phi_2 \mid \boldsymbol{x};\alpha_1;\alpha_2) = \frac{|\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)\cap\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}|\Phi_2;\alpha_2)|+|\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)\cap\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}|\Phi_2;\alpha_2)|}{|\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)|+|\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}|\Phi_1;\alpha_1)|}$, subject to $\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x} \mid \Phi_1;\alpha_1) = \{S \in \Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x} \mid \Phi_1;\alpha_1) : |S| = k\}$, and $\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x} \mid \Phi_1;\alpha_1) = \{S \in \Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x} \mid \Phi_1;\alpha_1) : |S| = k\}$. We set the saliency parameter $\alpha_1 = 0.2$ to select salient interactions. As shown in Fig. 11, low-order interactions exhibited higher consistency than high-order interactions. Notably, the consistency value of the salient first-order interactions encoded by the LLM Φ_1 is close to 1.0 on the ARC, SQuAD, and WikiQA data sets, while it is close to 0.7 on the BBC data set (which may stem from differences in data set difficulty). Results show that most of the salient first-order interactions can also be captured as salient interactions by the LLM Φ_2.

Fig. 12 also shows that **LLMs with superior generalization power are more likely to mainly encode low-order interactions and avoid learning high-order interactions**. That is, models with good generalization, such as the LLaMA-7B, mainly encode low-order interactions and rarely encode high-order interactions. In contrast, models with poor generalization, such as the BERT-base and the BERT-large (Devlin et al., 2018), tend to encode interactions across different orders.



Input sentence: Steam engines are external combustion engines, where the working fluid is separate from the combustion products. Non-combustion heat sources such as solar power, nuclear power or geothermal energy may be used. The ideal thermodynamic cycle used to analyze this process is called the
LLM's prediction token: Rank

Figure 10: Examples of consistent and inconsistent interactions encoded by the OPT-30B model and the LLaMA-7B model. Consistent interactions (represented by blue nodes) denote salient interactions shared by both the OPT-30B model and the LLaMA-7B model, while inconsistent interactions (represented by red nodes) denote salient interactions exclusively encoded by the OPT-30B model. Underlined words denote input variables selected to compute interactions[7].

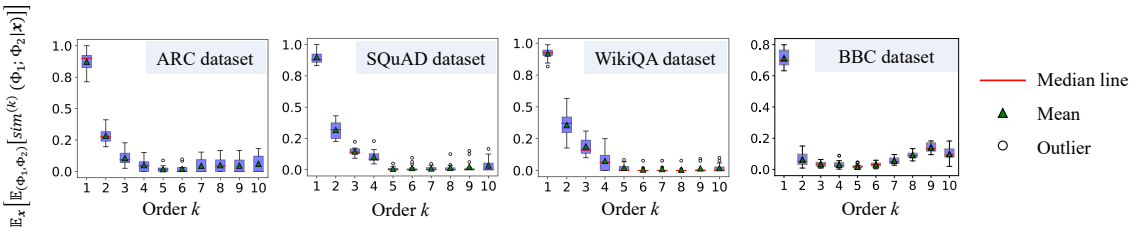


Figure 11: Boxplot of the consistency values across different orders. The consistency value of interactions extracted from a pair of LLMs $sim^{(k)}(\Phi_1;\Phi_2|\boldsymbol{x};\alpha_1;\alpha_2)$ is averaged over interactions of each specific order and over different input prompts. The Boxplot is computed based on the consistency values over different pairs of LLMs. Results show that low-order interactions exhibited higher consistency than high-order interactions.

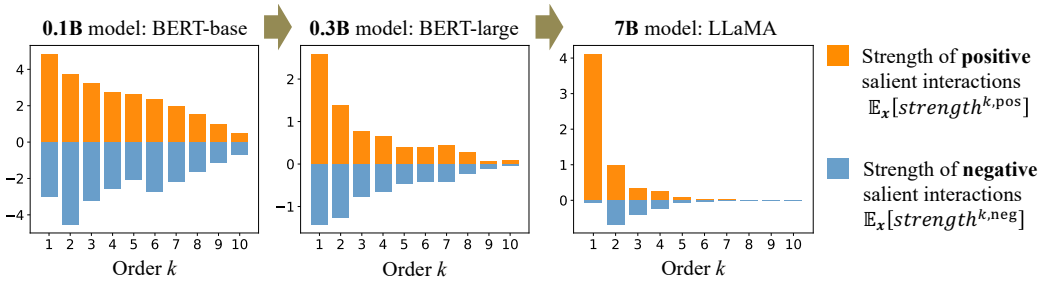


Figure 12: The distribution of salient positive interactions and salient negative interactions over different orders. Interaction strength is measured on the WikiQA data set. Results show that LLMs with superior generalization power are more likely to mainly encode low-order interactions and avoid learning high-order interactions.

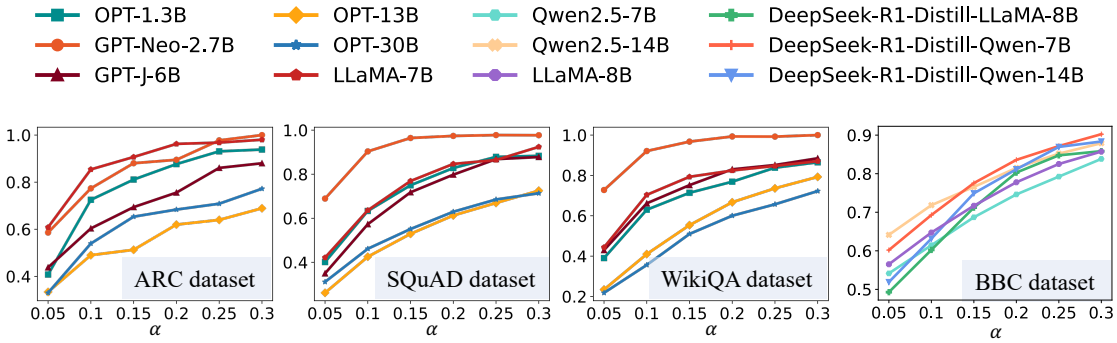


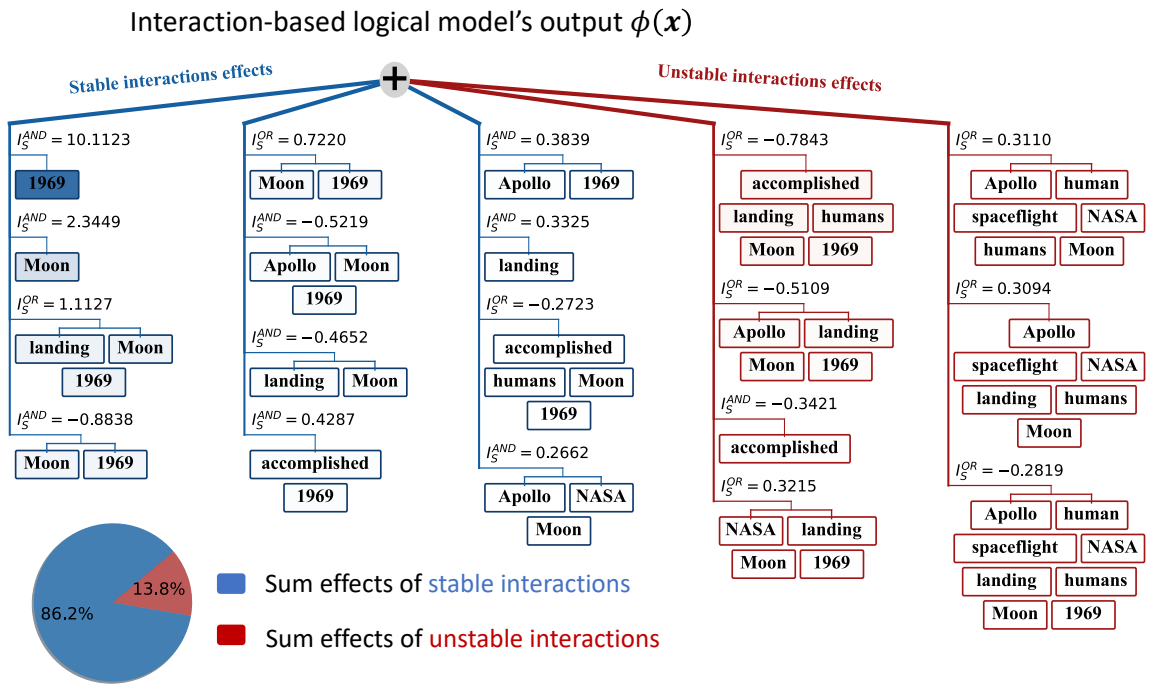
Figure 13: Stability of interactions $sim(\boldsymbol{x}^{(1)};\boldsymbol{x}^{(2)}|\Phi,\alpha_1,\alpha_2)$ across semantically similar sentences. Stability values are computed by setting different saliency parameters α_1. Results show that interactions of higher saliency usually have higher stability values across semantically similar sentences than not-so-salient interactions.

4.2 Stability of Interactions over Different Samples

Quantifying the stability of interactions is crucial for analyzing an LLM, *i.e.*, examining whether changing some semantically unrelated tokens in the prompt will change the extracted interactions. Given two semantically similar prompts $\boldsymbol{x}^{(1)}$ = "*Diabetes affects how the body processes*" and $\boldsymbol{x}^{(2)}$ = "*Diabetes affects the way the body uses*," the LLM Φ generates the same token $y^{\text{truth}} = [glucoses]$. If an interaction S has significant effects on both the inference of $\boldsymbol{x}^{(1)}$ and the inference of $\boldsymbol{x}^{(2)}$, then we can consider that the interaction S is stable across the two prompts. Otherwise, we think the interaction S is unstable.

Therefore, given a pair of semantically similar prompts $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$, we propose to quantify the stability of interactions extracted from $\boldsymbol{x}^{(1)}$ as the ratio of salient interactions extracted from $\boldsymbol{x}^{(1)}$ that can also be extracted from $\boldsymbol{x}^{(2)}$, *i.e.*, $sim(\boldsymbol{x}^{(1)};\boldsymbol{x}^{(2)}|\Phi,\alpha_1,\alpha_2) = \frac{|\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi,\alpha_1)\cap\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}^{(2)}|\Phi,\alpha_2)|+|\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi,\alpha_1)\cap\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}^{(2)}|\Phi,\alpha_2)|}{|\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi,\alpha_1)|+|\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi,\alpha_1)|}$. In experiments, we select prompts used in Section 3.2 as $\boldsymbol{x}^{(2)}$, and we shorten $\boldsymbol{x}^{(2)}$ to generate its corresponding $\boldsymbol{x}^{(1)}$, so as to generate a pair of prompts $(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)})$ for testing.[11] We try different saliency parameters $\alpha_1 \in [0.05, 0.3]$ to collect different sets of interactions $\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi,\alpha_1)$ and $\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi,\alpha_1)$

11. Please see Appendix F.4 for details about experimental data.



Input sentence: The Apollo program, was the third United States human spaceflight program carried out by (NASA), which accomplished landing the first humans on the Moon from 1969 to
LLM's prediction token: 1972

Figure 14: Examples of stable and unstable interactions extracted from the OPT-30B model. Stable interactions (represented by blue nodes) denote salient interactions that can be stably extracted from semantically similar sentences, whereas unstable interactions (represented by red nodes) denote salient interactions unique to the above input sentence. Underlined words denote input variables selected to compute interactions[7].

from $\boldsymbol{x}^{(1)}$, and set the saliency parameter $\alpha_2 = 0.05$ to collect two sets of interactions $\Omega^{\text{AND}}_{\text{salient}}(\boldsymbol{x}^{(2)}|\Phi, \alpha_2)$ and $\Omega^{\text{OR}}_{\text{salient}}(\boldsymbol{x}^{(2)}|\Phi, \alpha_2)$.

Fig. 13 shows that the stability of interactions increases along with the increase of the saliency parameter α_1. This implies that the more salient an interaction is, the more likely it is to remain stable across semantically similar prompts. When we set $\alpha_1 = 0.2$, on average, 77.38% of salient interactions remain stable across prompts. This also proves the relatively high stability of salient interactions through semantically similar prompts.

Fig. 14 visualizes the interaction patterns encoded by the OPT-30B model to compare stable interactions with unstable interactions. Results show that stable interactions tend to be simple and low-order, whereas complex, higher-order interactions often exhibit instability. This observation motivates further investigation into the stability of interactions across different orders. To this end, we use the following metric to measure the stability of interactions of the k-th order, *i.e.*,

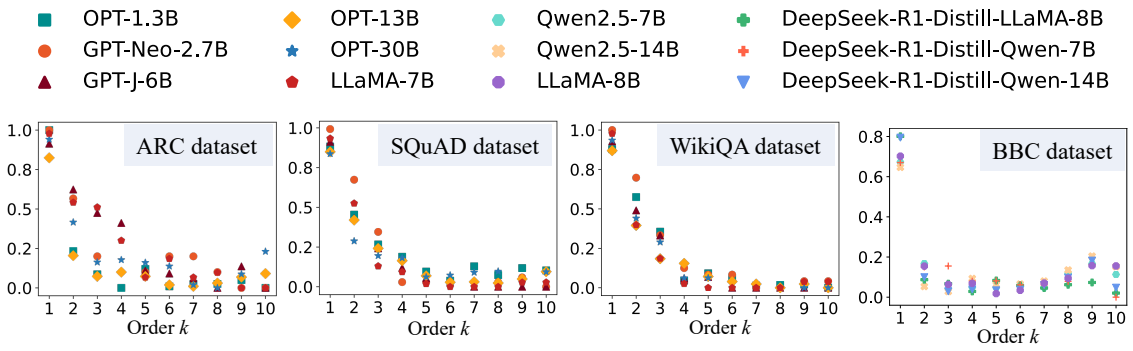


Figure 15: Stability of salient interactions of different orders $sim^{(k)}(\boldsymbol{x}^{(1)};\boldsymbol{x}^{(2)}|\Phi;\alpha_1;\alpha_2)$. The stability value is averaged over different input prompts. Results show that low-order interactions exhibited higher stability than high-order interactions.

$sim^{(k)}(\boldsymbol{x}^{(1)};\boldsymbol{x}^{(2)}|\Phi;\alpha_1;\alpha_2) = \frac{|\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi;\alpha_1)\cap\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}^{(2)}|\Phi;\alpha_2)|+|\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi;\alpha_1)\cap\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}^{(2)}|\Phi;\alpha_2)|}{|\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi;\alpha_1)|+|\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}^{(1)}|\Phi;\alpha_1)|}$.

We set the saliency parameter $\alpha_1 = 0.2$ to select salient interactions. Fig. 15 shows that low-order interactions exhibited higher stability than high-order interactions. In particular, the stability of the salient first-order interactions extracted from the prompt $\boldsymbol{x}^{(1)}$ is close to 1.0 on the ARC, SQuAD, and WikiQA data sets, while it is close to 0.7 on the BBC data set (may stem from different data set difficulty). This indicates that most of its salient first-order interactions can also be extracted as salient interactions from $\boldsymbol{x}^{(2)}$.

5. Application

In this section, we introduce three potential applications based on the interaction. First, we use the extracted interactions to diagnose the causes for LLMs' incorrect predictions, which helps reveal potential issues in their feature representations (see Section 5.1). Furthermore, we use the interaction's complexity as a new principle to help analyze the optimal pruning ratio for network compression (see Section 5.2). Finally, we use the interaction to evaluate the semantic soundness of the inference logic utilized by LLMs (see Section 5.3).

5.1 Exploring Defective Semantic Patterns in an LLM

LLMs commonly face issues such as hallucinations, data biases, and random responses. Prior research (Yang et al., 2023; Zhang et al., 2023; Hou et al., 2025) has primarily analyzed these issues through the lens of accuracy metrics, which often fail to uncover the underlying mechanisms that cause such issues. In this paper, we use interactions to reveal potential problematic feature representations that cause the LLM to generate incorrect answers.

Analyzing the hallucination problem. We first use interactions to diagnose hallucinations by examining the meaning of interaction patterns that support the incorrect answer, and Fig. 16 presents two case studies for illustration. In the first example, based on the input prompt $\boldsymbol{x}$ = "*Mary is the wife of John. James is the son of John. John is the son of*," the LLM is expected to answer "*unknown*." However, the OPT-1.3B model incorrectly

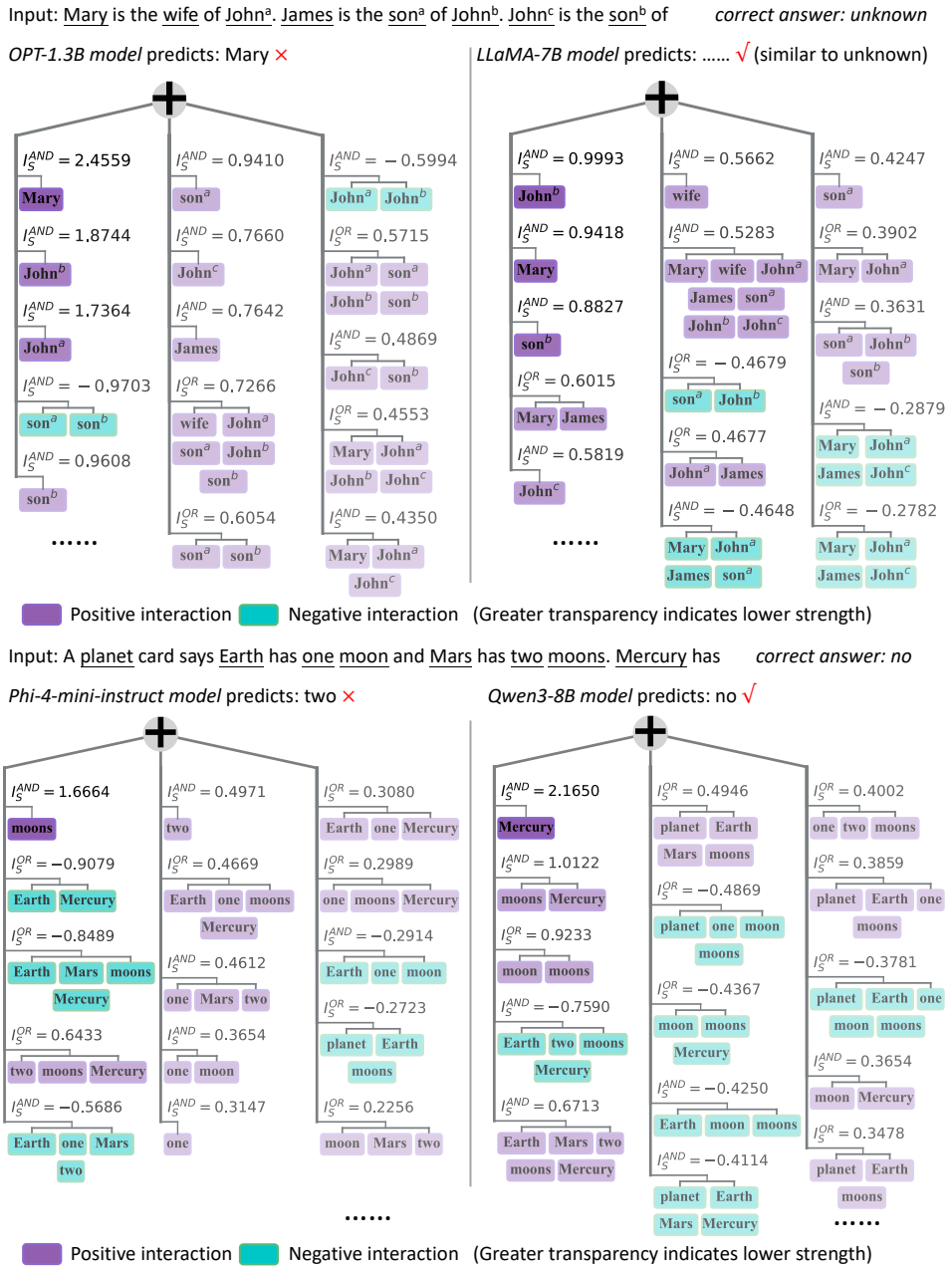


Figure 16: Using interactions to analyze hallucination patterns in two reasoning examples. (Top) Family-relation reasoning: OPT-1.3B predicts "*Mary*" (hallucination), while LLaMA-7B outputs "....." (correct answer, as no relevant information is provided by the prompt). (Bottom) Factual reasoning about planetary moons: Phi-4-mini-instruct predicts "*two*" (hallucination), whereas Qwen3-8B correctly predicts "*no*." Please see Appendix F.8 for more hallucination examples. Underlined words denote input variables selected to compute interactions[7]. Superscripts (a and b) disambiguate repeated words and are not input to the LLM.

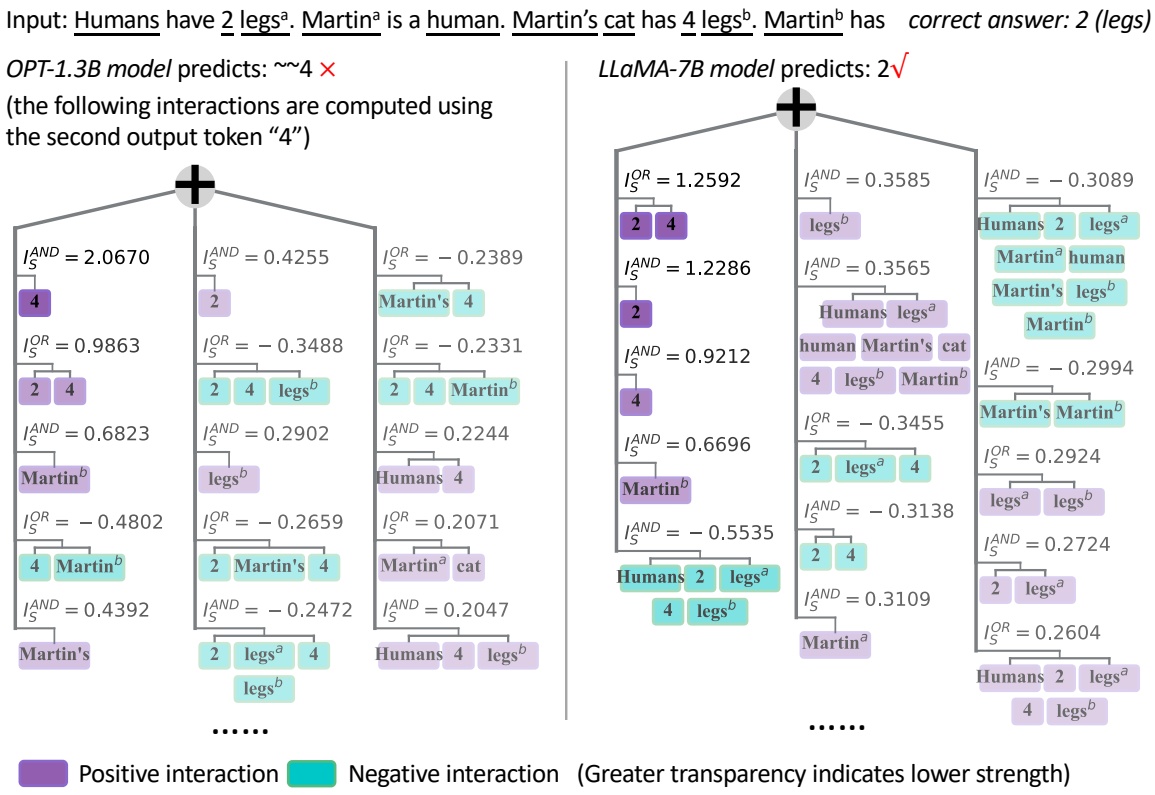


Figure 17: An example of using interactions to analyze why LLMs' predictions are affected by irrelevant facts in the input. Underlined words denote input variables selected to compute interactions[7]. Superscripts (a and b) disambiguate repeated words and are not input to the LLM.

predicts "*Mary*," which is a hallucination problem. In comparison, the LLaMA-7B model does not exhibit this issue, *i.e.*, it outputs "....." for $\boldsymbol{x}$, which is closer to the correct answer "*unknown*." As shown in Fig. 16 (top), the OPT-1.3B model incorrectly encodes a strong prior that Mary is the mother of John, because an incorrect interaction $S =$ "*Mary*" boosts the generation confidence of "*Mary*" by $I_S^{AND} = 2.4559$. This is a partial reason behind the hallucination problem. In contrast, although the LLaMA-7B model also encodes such an incorrect prior interaction, the interaction effect is much weaker than that in the OPT-1.3B model. The lack of dominating interactions leads to a more acceptable answer (*i.e.*, ".....").

In the second case, given the prompt $\boldsymbol{x} =$ "*A planet card says Earth has one moon and Mars has two moons. Mercury has*," the correct answer is "*no moon.*" However, as shown in Fig. 16 (bottom), the Phi-4-mini-instruct model (Microsoft et al., 2025) incorrectly predicts "*two.*" The extracted interactions reveal that this hallucination stems primarily from the model's over-reliance on the specific pattern "*Mars has two moons.*" In particular, salient positive interactions involving "*moons*," "*two, Mercury*," and "*moons, Mercury*" support the generation of "*two*," indicating that the model mistakenly transfers the Mars-related fact to Mercury. In contrast, the Qwen3-8B model (Yang et al., 2025) correctly predicts the token "*no.*" Its salient interactions focus more on Mercury, suggesting that Qwen3-8B

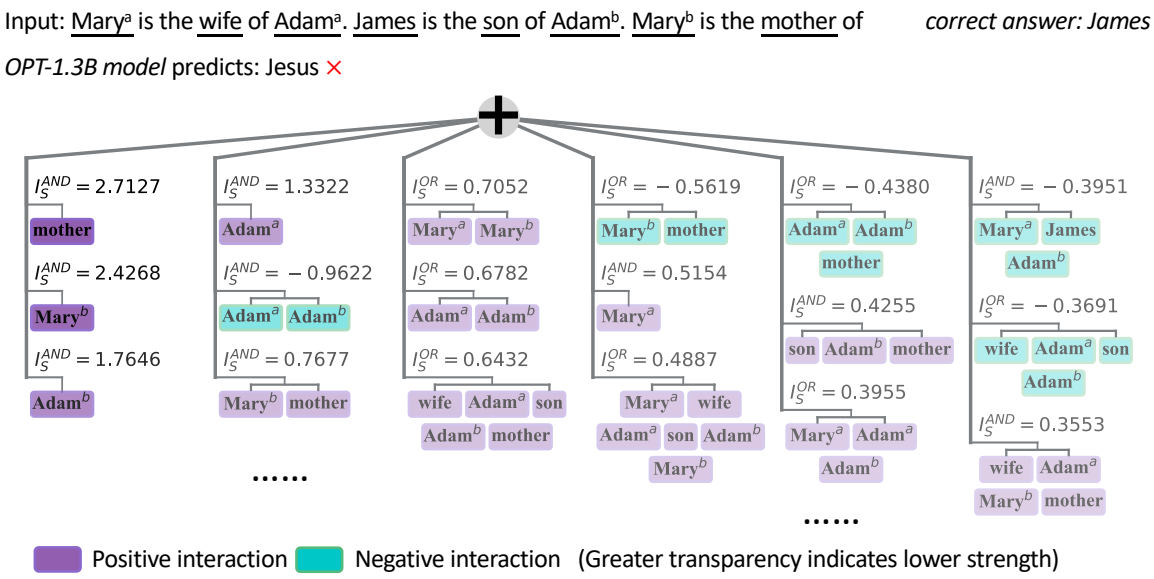


Figure 18: An example for data bias. We select meaningful words (underlined) from each input sentence as input variables to compute interactions to reduce the computational cost[7]. Superscripts (a and b) disambiguate repeated words and are not input to the LLM.

better distinguishes the queried entity Mercury from the surrounding facts about Earth and Mars. Please see Appendix F.8 for additional hallucination cases.

Analyzing why LLMs' predictions are affected by irrelevant facts in the input. Fig. 17 shows an example where a distracting sentence ("*Martin's cat has 4 legs.*") makes the OPT-1.3B model make an incorrect prediction of "4" (the correct answer should be "2"). Results show that the OPT-1.3B model has imprudently learned the interaction $S = \{4\}$ with a dominating effect on the number of legs, which leads to the incorrect prediction of "4." In comparison, the LLaMA-7B model predicts the correct answer of "2." We observe that the LLaMA-7B model encodes an interaction $S = \{Humans, legs^a, human, Martin's, cat, 4, legs^b, Martin^b\}$, which indicates that the LLaMA-7B model successfully encodes the explicit knowledge that the human beings have 2 legs. Superscripts a and b are used to distinguish the same word in two different positions, which are not input to the LLM. Therefore, the LLaMA-7B model correctly answers the fact that Martin, as a "human," has 2 legs. Whereas, the OPT-1.3B model does not encode such prior knowledge.

Analyzing how predictions of an LLM are influenced by the bias of the training data. As Fig. 18 shows, given the input sentence "*Mary is the wife of Adam. James is the son of Adam. Mary is the mother of*," the correct prediction of the next token should be "*James*." However, the OPT-1.3B model predicts the incorrect answer of "*Jesus*." It might be because the OPT-1.3B model overly relies on the information about Christianity and related figures in training samples. Therefore, when the OPT-1.3B model sees "*Mary*" and "*Adam*," it tends to associate them with "*Jesus*," and fails to encode the exact correlation between the three terms "*Mary*," "*wife*," and "*Adam*." From the perspective of interactions, although the top-10 AND-OR interactions encoded by the OPT-1.3B model include "*Mary*" or "*Adam*," no interaction simultaneously contains the

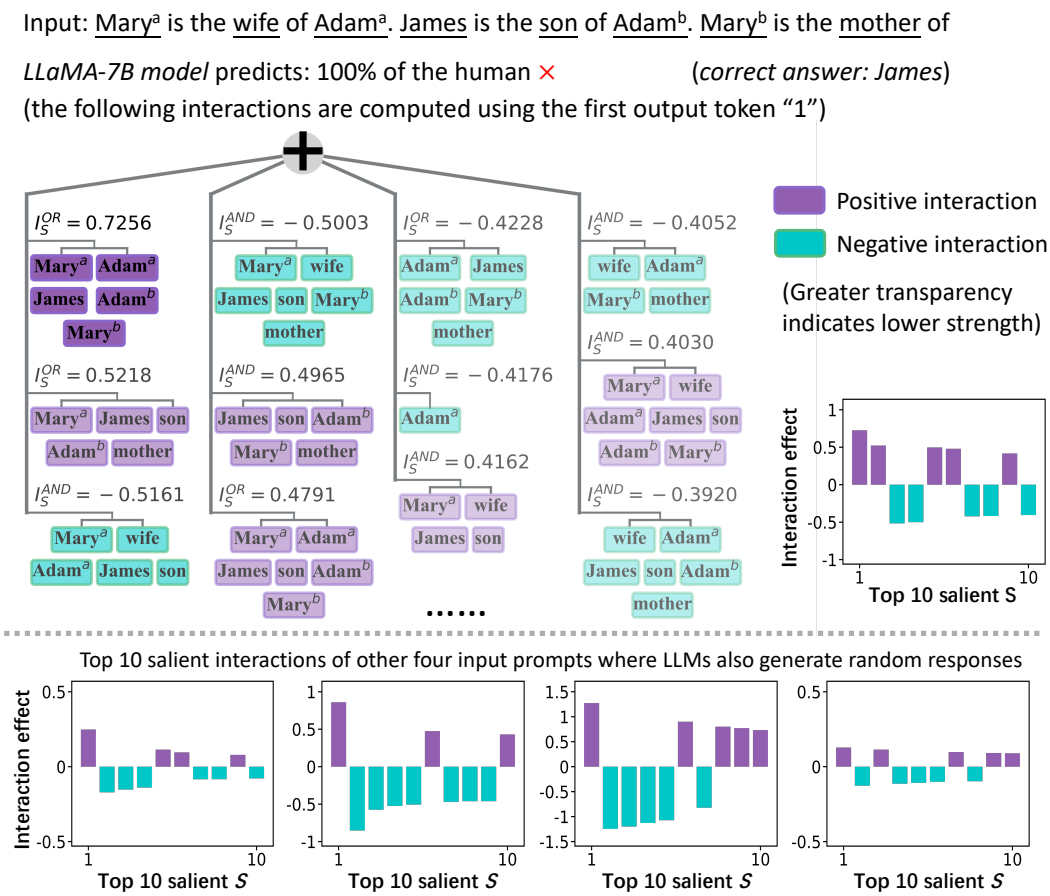


Figure 19: Visualizing the top-ranked salient interactions when LLMs generate random responses. Each bar chart displays the mutually canceling numerical effects of the top 10 salient interactions, which are extracted from a specific input prompt. Detailed interactions on the first prompt are shown above, and detailed interactions for three additional prompts are provided in Fig. 25 in Appendix F.7. Underlined words denote input variables selected to compute interactions[7]. Superscripts (a and b) disambiguate repeated words and are not input to the LLM.

three terms "*Mary*," "*wife*," and "*Adam*" as a single inference pattern, which validates the above claim, to some extent.

Analyzing the phenomenon of random responses. We observe that sometimes a significant portion of salient interaction effects cancel each other out when the LLM generates random responses. As Fig. 19 shows, given the input sentence "*Mary is the wife of Adam. James is the son of Adam. Mary is the mother of*," the correct prediction of the next token should be "*James*." However, the LLaMA-7B model provides a random response of "*100% of the human*." From the perspective of interactions, the top 10 AND-OR interactions extracted from the input sentence *w.r.t.* the generation of the first token ("*100*") are mostly complex (high-order) interactions, and most of these interaction effects cancel each other out. The bar chart in Fig. 19 visualizes the effect values of the top 10

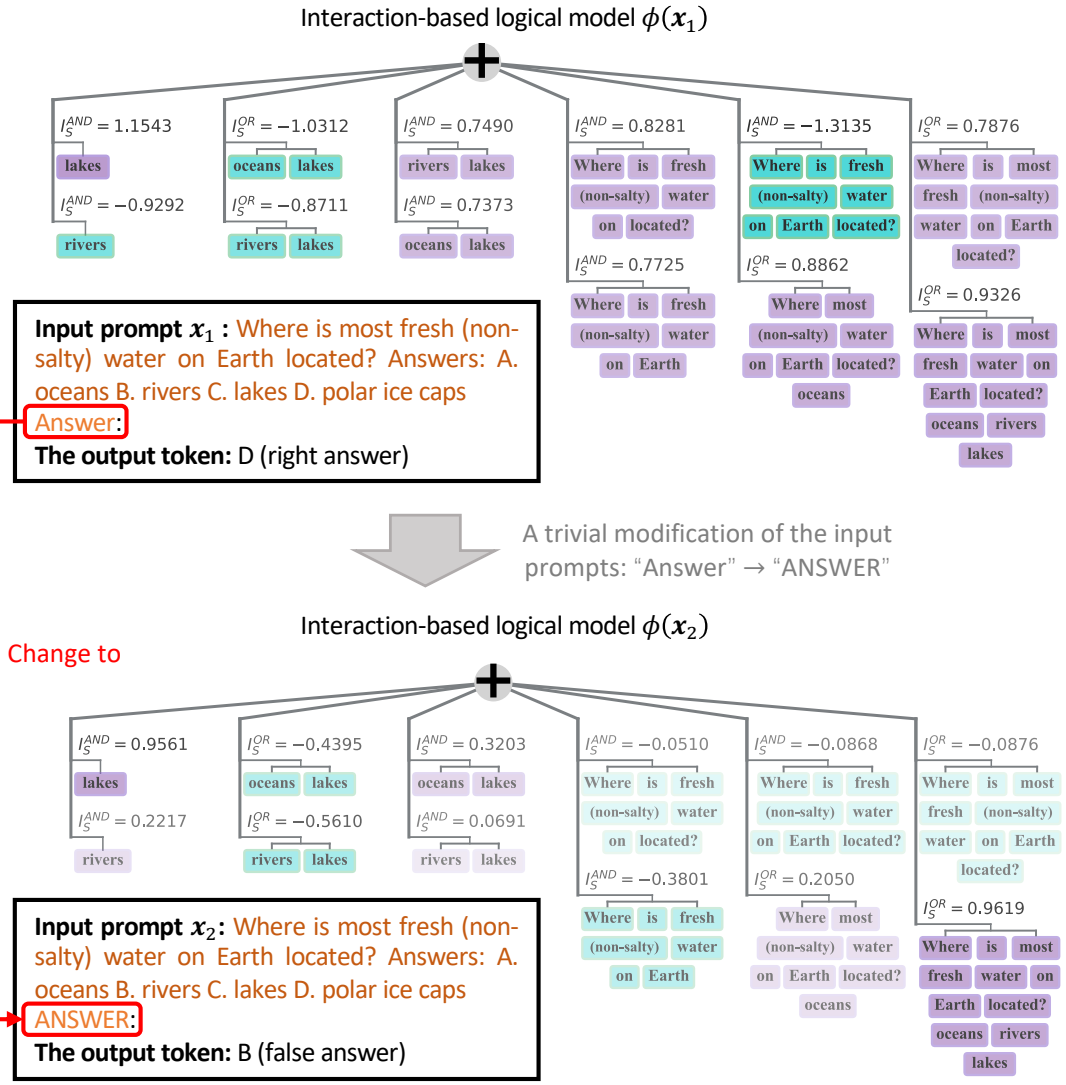


Figure 20: Comparing the top 12 salient interactions extracted from the original prompt with the same interactions extracted from a slightly revised prompt. Results show that a trivial prompt modification ("*Answer*" → "*ANSWER*") significantly changes the effects of the 12 interactions, particularly high-order interactions. Since the LLM's output score on this input prompt heavily relies on high-order and highly sensitive interactions, the model's overall output becomes vulnerable to minor prompt perturbations.

salient interactions. We observe that it exhibits alternating positive and negative values with similar magnitudes, which leads to near-total cancellation. Such cancellation between positive and negative interactions reflects a breakdown in the reasoning process of the LLaMA-7B model. Besides, Fig. 19 also includes bar charts of interactions extracted from four additional samples in which LLMs also generate random responses. They also exhibit the cancellation phenomenon. Detailed interaction results for three additional samples are provided in Fig. 25 in Appendix F.7.

Analyzing the prompt sensitivity. Prompt sensitivity reflects the instability of an LLM's outputs, where specific changes of some semantically irrelevant tokens may lead to significant changes in the LLM's output. As shown in Fig. 20, the original input prompt is $\boldsymbol{x}_1$ = "*Where is most fresh (non-salty) water on Earth located? Answers: A. oceans B. rivers C. lakes D. polar ice caps Answer:.*" Changing "*Answer*" to "*ANSWER*" causes the LLaMA-7B model (Touvron et al., 2023a) to switch from the correct answer "*D*" to the incorrect answer "*B*." Results in Fig. 20 reveal that such a trivial modification of the input prompts can cause a complete shift in both the model's generated token and the effects of its top salient interactions. In particular, high-order interactions exhibit strong sensitivity to such input modification. Since the LLM's output score heavily relies on these high-order and highly sensitive interactions for this input prompt, the model's overall output becomes vulnerable to minor prompt perturbations. In fact, such sensitivity of high-order interactions has already been illustrated in Section 4.2 (see Fig. 15). It is the high sensitivity of high-order interactions that makes the entire LLM's inference sometimes sensitive to even subtle changes in the input prompt.

5.2 Helping Analyze the Optimal Pruning Ratio for Network Compression

In this subsection, we are inspired by Li et al. (2026) and use interactions to analyze and assist in selecting the optimal pruning ratio for network compression. The pruning ratio corresponds to the proportion of neurons removed from the neural network. A higher pruning ratio implies that a greater proportion of neurons are removed, which can substantially degrade model performance. Therefore, it is crucial to choose an appropriate pruning ratio. Network compression typically seeks to remove as many neurons as possible while preserving model performance. In this work, we empirically find that the interaction can effectively elucidate the effect of the pruning ratio on model performance, thereby offering a new method for pruning ratio selection. We conduct pruning experiments on LLaMA-7B, Llama-3.1-8B, and DeepSeek-R1-Distill-Llama-8B using the Wanda pruning method (Sun et al., 2023), and extract the interactions encoded by the pruned models. The data set consists of one hundred sentences from The Economist and government work reports.

Pruning neurons ultimately removes interactions encoded by the LLM. Fig. 21 shows changes in the distributions of interactions when the pruning ratio increases. Results show that as the pruning ratio increases, (1) the accuracy declines moderately at first, and (2) then declines rapidly later. Interaction analysis reveals that in the first phase, the network compression method primarily removes interactions of medium and high orders. Then, in the second phase when the pruning ratio reaches 40.0%, the strength of almost all mid-order and high-order interactions approaches zero. In contrast, the strength of low-order interactions remains relatively high. As the pruning ratio continues to rise, the network compression method further removes both low-order and high-order interactions. These results indicate that removing medium and high-order interactions has a comparatively smaller impact on accuracy than removing low-order interactions. This can be attributed to the fact that high-order interactions often correspond to non-generalizable and noisy patterns, while low-order interactions are essential for core and generalizable reasoning. This interaction-based perspective provides a new principle to analyze the effects of network compression on interaction patterns.

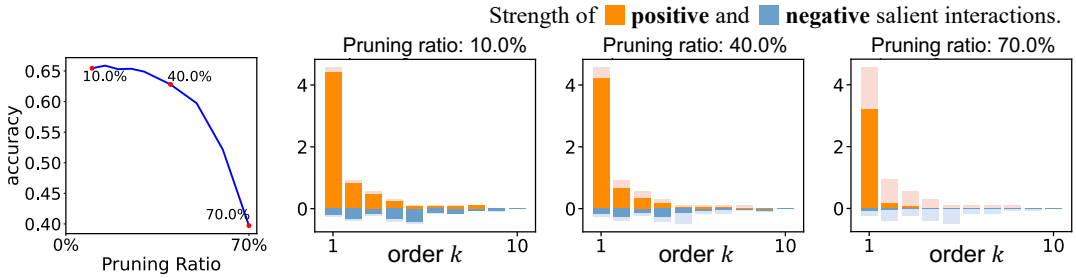


Figure 21: Evolution of interaction distributions under increasing pruning ratios for network compression. Results show that accuracy declines first moderately, then more rapidly. Given a small pruning ratio, the LLM preferentially reduces medium and high-order interactions. The accuracy drops sharply only when low-order interactions are compromised under a large pruning ratio. Thus, such low-order interactions provide new insights into network compression quality.

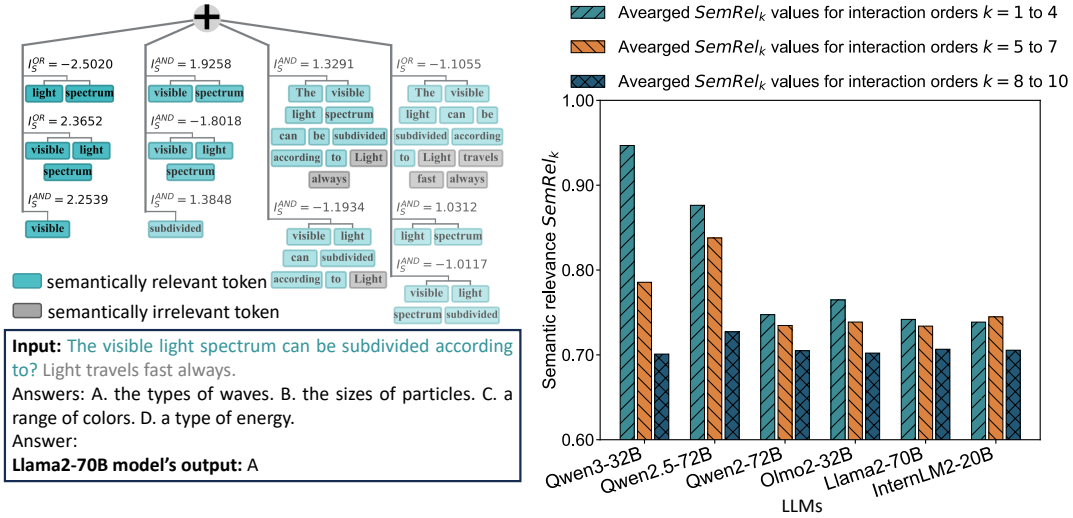


Figure 22: (**Left**) Illustrating interactions exclusively containing semantically relevant tokens and those containing semantically irrelevant tokens. (**Right**) Comparing semantic relevance between low-order interactions, middle-order interactions, and high-order interactions. Results show that low-order interactions usually exhibit stronger semantic relevance than high-order interactions.

5.3 Evaluating Semantic Relevance of Inference Logics via Interactions

In this subsection, we use interactions to evaluate the semantic relevance of LLMs' inference logics, *i.e.*, the proportion of interactions that focus on tokens relevant to the target output. To quantify the semantic relevance of interactions, we classify input variables into two sets: (1) **a set of relevant input variables**, which are denoted by R and can be considered as true causal factors for the target output, and (2) **a set of irrelevant input variables**, which are logically irrelevant for the target output. For example, given the input prompt "*Humans have 2 legs. Martin is a human. Martin's cat has 4 legs. Martin has*," an LLM is expected to generate "*2 legs.*" In this case, only input variables in

$R = \{Humans, have, 2, legs, Martin, is, a, human\}$ are relevant to the target output "*2 legs.*" The remaining input variables in $N \setminus R$ are irrelevant to the output. Then, we define the semantic relevance of an interaction S as the ratio of relevant input variables that S comprises, as follows.

$$SemRel(S) = \frac{|S \cap R|}{|S|}. \tag{12}$$

The original input prompt contains both relevant input variables and some background input variables that are not very relevant to the target output. Besides, to address cases where irrelevant variables are too sparse in certain prompts, we can optionally insert a sentence irrelevant to the target output into each input prompt. In this way, the inserted words can be annotated as definitively irrelevant input variables. For example, given the prompt "*Which characteristic describes the texture of a kitten's fur? Answers: A. gray B. warm C. long D. soft Answer:*," we insert an irrelevant sentence "*Kittens have four legs*" at the very beginning of the given prompt. In this case, relevant input variables are annotated as $R = \{Which, characteristic, describes, the, texture, of, a, kitten's, fur\}$, and irrelevant input variables are $\{Kittens, have, four, legs\}$. The remaining input variables in the prompt, *i.e.*, "*Answers: A. gray B. warm C. long D. soft Answer:*," are considered as background and are excluded from the computation of interactions.

Then, the average semantic relevance of all the k-th order interactions is quantified as follows.

$$SemRel_k = \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{S\in\Omega^{\text{AND},k}_{\text{salient}}(\boldsymbol{x}|\alpha)\cup\Omega^{\text{OR},k}_{\text{salient}}(\boldsymbol{x}|\alpha)} SemRel(S)\right]. \tag{13}$$

We conduct experiments to evaluate the semantic relevance of six LLMs, including the Llama2-70B model (Touvron et al., 2023b), the Qwen2-72B model (Team et al., 2024), the Qwen2.5-72B model (Qwen et al., 2024), the Qwen3-32B model (Yang et al., 2025), the OLMo2-32B model (OLMo et al., 2025), and the InternLM2-20B model (Cai et al., 2024). We insert irrelevant sentences to the multiple-choice questions of the ARC data set (Clark et al., 2018) to obtain testing samples for evaluating the semantic relevance of these six LLMs. As Fig. 22 shows, across all LLMs, the semantic relevance of low-order interactions is consistently higher than that of medium-order interactions and high-order interactions. Notably, the Qwen2.5-72B model and the Qwen3-32B model exhibit significantly greater semantic relevance of low-order interactions compared to other LLMs, with semantic relevance values ranging between 0.8 and 1.0.

6. Conclusion and Discussions

This study proposes an interaction-based framework for analyzing and interpreting LLMs. We first empirically demonstrate that the outputs of an LLM can be represented as a concise logical model based on a small number of primitive interactions. Building on this, we discover that the interaction metric can be used to probe the internal mechanisms of LLMs from two perspectives. First, we extend the interaction metric to disentangle interaction patterns for coarse-grained inference and interaction patterns for fine-grained inference. We empirically find that LLMs tend to rely on low-order interactions for coarse-grained inference and employ more high-order interactions for fine-grained inference. Second, we extend the interaction metric to a layer-wise analysis, *i.e.*, we track the evolution of interaction patterns through layers in LLMs during forward propagation. Experimental results show that as

depth increases, LLMs progressively learn more interactions, with an increasing proportion of high-order interactions. This indicates that with the forward propagation process, LLMs progressively construct increasingly complex logical representations.

We discover that the interaction-based framework can also be extended to evaluate the representation quality of LLMs and to explore defective semantic patterns in LLMs. In terms of evaluating the representation quality of LLMs, we introduce interaction-based metrics for evaluating consistency of interactions over LLMs and stability of interactions over different samples. Based on these metrics, we observe that given the same input prompt, different LLMs share a set of optimal interactions. Additionally, a single LLM exhibits high stability in the interactions it captures across semantically similar prompts. All these findings indicate that the extracted interactions can be regarded as faithful primitive inference patterns used by the LLM. Furthermore, the interaction patterns can be considered to determine the performance. We empirically find that LLMs with better generalization tend to encode a higher proportion of low-order interactions.

In terms of exploring defective semantic patterns in LLMs, we find that the interaction successfully reveals the causes of several typical prediction errors. Specifically, LLMs may learn and overly rely on interactions stemming from statistical biases in the training data. These interactions can be viewed as reflecting incorrect yet ingrained strong associations, and such strong associations can override rational reasoning based on the current input prompt, even when contextual information is contradictory.

We also use interactions to help analyze and guide network compression task. We empirically observe a strong correlation between the low-order interactions and model performance. This correlation emphasizes the potential of using low-order interactions as a metric to help determine the optimal pruning ratio.

Moreover, we investigate the semantic relevance of the interactions used by the LLM to generate the target output. We quantify the proportion of semantically relevant interactions, so as to evaluate the reliability of the internal decision logic of a neural network.

A limitation of our current study is the high computational cost associated with the interaction framework. Specifically, according to the interaction formula given in Theorem 1, computing the interactions modeled on a single sample requires 2^n inference operations. Fortunately, we still have a practical method to reduce the computational cost. When applying Integrated Gradients to visualize attributions across input tokens, we observe that the tokens most relevant to the target output are often concentrated within 5 to 10 contiguous textual regions. By merging the tokens within each such region into a single phrase and treating it as an aggregated input variable, we can compute interactions in a substantially more efficient manner.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 92370115, 62276165, and 62576249), the Shanghai Natural Science Foundation (No. 24ZR1491700), the Shanghai Science and Technology Commission (No. 25511102900), and the Shanghai Municipal Education Commission (No. 24CGA20). The authors declare that they have no competing interests.

Appendix A. Strategies of Masking Input Variables

In the research of attribution methods, it is common to use a specific embedding to mask input variables of a DNN (Lundberg and Lee, 2017; Ancona et al., 2019; Fong et al., 2019), and use changes in network outputs on the masked samples to estimate attributions of input variables. However, each method has certain limitations. The mean baseline value (Dabkowski and Gal, 2017), that is, setting the baseline value for each input variable to its mean across all samples, introduced additional signals, *e.g.*, grey dots in images. Similarly, the zero baseline value (Ancona et al., 2019; Sundararajan et al., 2017), that is, setting baseline values for all input variables to zero, would also introduce additional signals to the input, such as black dots. Additionally, blurring image pixels using a Gaussian kernel (Fong and Vedaldi, 2017; Fong et al., 2019) as the masked state removes high-frequency signals but fails to eliminate low-frequency signals (Covert et al., 2020; Sturmfels et al., 2020).

Appendix B. Mathematical Proof of the Universal Matching Property in Theorem (1)

We follow Chen et al. (2024); Zhou et al. (2023a) to prove the universal matching property as follows.

Proof (1) Universal matching property of AND interactions.

We prove that the output component u_A^{AND} on all 2^n masked samples $\{\boldsymbol{x}_A : A \subseteq N\}$ could be universally explained by all AND interactions contained in A, *i.e.*, $\forall \emptyset \neq A \subseteq N, u_A^{\text{AND}} = \sum_{\emptyset \neq S \subseteq A} I_S^{\text{AND}} + v(\boldsymbol{x}_\emptyset)$. In particular, we define $u_\emptyset^{\text{AND}} = v(\boldsymbol{x}_\emptyset)$ (*i.e.*, we attribute output on an empty sample to AND interactions).

Specifically, the AND interaction is defined as $I_S^{\text{AND}} = \sum_{L \subseteq S} (-1)^{|S|-|L|} u_L^{\text{AND}}$ in Theorem 1. To compute the sum of AND interactions $\sum_{\emptyset \neq S \subseteq A} I_S^{\text{AND}} = \sum_{\emptyset \neq S \subseteq A} \sum_{L \subseteq S} (-1)^{|S|-|L|} u_L^{\text{AND}}$, we first exchange the order of summation of the set $L \subseteq S \subseteq A$ and the set $S \supseteq L$. That is, we compute all linear combinations of all sets S containing L with respect to the model outputs u_L^{AND} given a set of input variables L, *i.e.*, $\sum_{S: L \subseteq S \subseteq A} (-1)^{|S|-|L|} u_L^{\text{AND}}$. Then, we compute all summations over the set $L \subseteq A$.

In this way, we can compute them separately for different cases of $L \subseteq S \subseteq A$. In the following, we consider the cases (1) $L = S = A$, and (2) $L \subseteq S \subseteq A, L \neq A$, respectively.

Case 1: $L = S = A$, the linear combination of all subsets S containing L with respect to the model output u_L^{AND} is $(-1)^{|A|-|A|} u_L^{\text{AND}} = u_L^{\text{AND}}$.

Case 2: $L \subseteq S \subseteq A, L \neq A$, the linear combination of all subsets S containing L with respect to the model output u_L^{AND} is $\sum_{S: L \subseteq S \subseteq A} (-1)^{|S|-|L|} u_L^{\text{AND}}$. For all sets $S : A \supseteq S \supseteq L$, let us consider the linear combinations of all sets S with number $|S|$ for the model output u_L^{AND}, respectively. Let $m := |S| - |L|$, $(0 \leq m \leq |A| - |L|)$, then there are a total of $C_{|A|-|L|}^m$ combinations of all sets S of order $|S|$. Thus, given L, accumulating the model outputs u_L^{AND} corresponding to all $S \supseteq L$, then $\sum_{S: L \subseteq S \subseteq A} (-1)^{|S|-|L|} u_L^{\text{AND}} = u_L^{\text{AND}} \cdot \underbrace{\sum_{m=0}^{|A|-|L|} C_{|A|-|L|}^m (-1)^m}_{=0} = 0$. Please see the complete derivation of the following formula.

$$
\begin{aligned}
&\sum\nolimits_{\emptyset\neq S\subseteq A} I_S^{\text{AND}} \\
=&\sum\nolimits_{\emptyset\neq S\subseteq A}\sum\nolimits_{L\subseteq S}(-1)^{|S|-|L|}u_L^{\text{AND}} \\
=&\sum\nolimits_{L\subseteq A}\sum\nolimits_{S:L\subseteq S\subseteq A}(-1)^{|S|-|L|}u_L^{\text{AND}} - u_\emptyset^{\text{AND}} \\
=&\underbrace{u_A^{\text{AND}}}_{L=A} + \sum\nolimits_{L\subseteq A, L\neq A} u_L^{\text{AND}} \cdot \underbrace{\sum\nolimits_{m=0}^{|A|-|L|} C_{|A|-|L|}^m (-1)^m}_{=0} - u_\emptyset^{\text{AND}} \\
=&u_A^{\text{AND}} - u_\emptyset^{\text{AND}} \\
=&u_A^{\text{AND}} - v(\boldsymbol{x}_\emptyset).
\end{aligned}
\tag{14}
$$

Thus, we have $\forall \emptyset \neq A \subseteq N, u_A^{\text{AND}} = \sum_{\emptyset\neq S\subseteq A} I_S^{\text{AND}} + v(\boldsymbol{x}_\emptyset)$.

(2) Universal matching property of OR interactions.

According to the definition of OR interactions, we will derive that $\forall A \subseteq N, u_A^{\text{OR}} = \sum_{S:S\cap A\neq\emptyset} I_S^{\text{OR}}$, where we define $u_\emptyset^{\text{OR}} = 0$ (recall that in Step (1), we attribute the output on empty input to AND interactions).

Specifically, the OR interaction is defined as $I_S^{\text{OR}} = -\sum_{L\subseteq S}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}}$ in Theorem 1. Similar to the above derivation of the Universal matching property of AND interactions, to compute the sum of OR interactions $\sum_{S:S\cap A\neq\emptyset} I_S^{\text{OR}} = \sum_{S:S\cap A\neq\emptyset}$ $\left[-\sum_{L\subseteq S}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}}\right]$, we first exchange the order of summation of the set $L \subseteq S \subseteq N$ and the set $S : S\cap A \neq \emptyset$. That is, we compute all linear combinations of all sets S containing L with respect to the model outputs $u_{N\setminus L}^{\text{OR}}$ given a set of input variables L, *i.e.*, $\sum_{S:S\cap A\neq\emptyset, S\supseteq L}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}}$. Then, we compute all summations over the set $L \subseteq N$.

In this way, we can compute them separately for different cases of $L \subseteq S \subseteq N, S\cap A \neq \emptyset$. In the following, we consider the cases (1) $L = N \setminus A$, (2) $L = N$, (3) $L \cap A \neq \emptyset, L \neq N$, and (4) $L \cap A = \emptyset, L \neq N \setminus A$, respectively.

Case 1: $L = N \setminus A$, the linear combination of all subsets S containing L with respect to the model output $u_{N\setminus L}^{\text{OR}}$ is $\sum_{S:S\cap A\neq\emptyset, S\supseteq L}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}} = \sum_{S:S\cap A\neq\emptyset, S\supseteq L}(-1)^{|S|-|L|}u_A^{\text{OR}}$. For all sets $S : S \supseteq L, S\cap A \neq \emptyset$ (then $S \neq N \setminus A, S \neq L$), let us consider the linear combinations of all sets S with number $|S|$ for the model output u_A^{OR}, respectively. Let $|S'| := |S| - |L|$, $(1 \leq |S'| \leq |A|)$, then there are a total of $C_{|A|}^{|S'|}$ combinations of all sets S' of order $|S'|$. Thus, given L, accumulating the model outputs u_A^{OR} corresponding to all $S \supseteq L$, then $\sum_{S:S\cap A\neq\emptyset, S\supseteq L}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}} = u_A^{\text{OR}} \cdot \underbrace{\sum_{|S'|=1}^{|A|} C_{|A|}^{|S'|}(-1)^{|S'|}}_{=-1} = -u_A^{\text{OR}}$.

Case 2: $L = N$ (then $S = N$), the linear combination of all subsets S containing L with respect to the model output $u_{N\setminus L}^{\text{OR}}$ is $\sum_{S:S\cap A\neq\emptyset, S\supseteq L}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}} = (-1)^{|N|-|N|}u_\emptyset^{\text{OR}} = u_\emptyset^{\text{OR}}$.

Case 3: $L \cap A \neq \emptyset, L \neq N$, the linear combination of all subsets S containing L with respect to the model output $u_{N\setminus L}^{\text{OR}}$ is $\sum_{S:S\cap A\neq\emptyset, S\supseteq L}(-1)^{|S|-|L|}u_{N\setminus L}^{\text{OR}}$. Let us split $|S| - |L|$ into $|S'|$ and $|S''|$, *i.e.*, $|S| - |L| = |S'| + |S''|$, where $S' = \{i|i \in S, i \notin L, i \in N \setminus A\}$, $S'' = \{i|i \in S, i \notin L, i \in A\}$ (then $0 \leq |S''| \leq |A| - |A \cap L|$). Thus, given L, accumulating

the model outputs $u^{\text{OR}}_{N\setminus L}$ corresponding to all $S \supseteq L$, then $\sum_{S:S\cap A\neq\emptyset,S\supseteq L}(-1)^{|S|-|L|}u^{\text{OR}}_{N\setminus L} = u^{\text{OR}}_{N\setminus L}\sum_{S'\subseteq N\setminus A\setminus L}\underbrace{\sum_{|S''|=0}^{|A|-|A\cap L|}C^{|S''|}_{|A|-|A\cap L|}(-1)^{|S'|+|S''|}}_{=0} = 0$.

Case 4: $L\cap A=\emptyset, L\neq N\setminus A$, the linear combination of all subsets S containing L with respect to the model output $u^{\text{OR}}_{N\setminus L}$ is $\sum_{S:S\cap A\neq\emptyset,S\supseteq L}(-1)^{|S|-|L|}u^{\text{OR}}_{N\setminus L}$. Similarly, let us split $|S|-|L|$ into $|S'|$ and $|S''|$, *i.e.*, $|S|-|L| = |S'|+|S''|$, where $S' = \{i|i\in S, i\notin L, i\in N\setminus A\}$, $S'' = \{i|i\in S, i\in A\}$ (then $0\leq|S''|\leq|A|$). Thus, given L, accumulating the model outputs $u^{\text{OR}}_{N\setminus L}$ corresponding to all $S\supseteq L$, then $\sum_{S:S\cap A\neq\emptyset,S\supseteq L}(-1)^{|S|-|L|}u^{\text{OR}}_{N\setminus L} = u^{\text{OR}}_{N\setminus L}\cdot\sum_{S'\subseteq N\setminus A\setminus L}\underbrace{\sum_{|S''|=0}^{|A|}C^{|S''|}_{|A|}(-1)^{|S'|+|S''|}}_{=0} = 0$.

Please see the complete derivation of the following formula.

$$
\begin{aligned}
\sum\nolimits_{S:S\cap A\neq\emptyset} I^{\text{OR}}_S &= \sum\nolimits_{S:S\cap A\neq\emptyset}\Big[-\sum\nolimits_{L\subseteq S}(-1)^{|S|-|L|}u^{\text{OR}}_{N\setminus L}\Big] \\
&= -\sum\nolimits_{L\subseteq N}\sum\nolimits_{S:S\cap A\neq\emptyset,S\supseteq L}(-1)^{|S|-|L|}u^{\text{OR}}_{N\setminus L} \\
&= -\left[\sum_{|S'|=1}^{|A|}C^{|S'|}_{|A|}(-1)^{|S'|}\right]\cdot\underbrace{u^{\text{OR}}_A}_{L=N\setminus A} - \underbrace{u^{\text{OR}}_\emptyset}_{L=N} \\
&\quad - \sum_{L\cap A\neq\emptyset,L\neq N}\left[\sum_{S'\subseteq N\setminus A\setminus L}\left(\sum_{|S''|=0}^{|A|-|A\cap L|}C^{|S''|}_{|A|-|A\cap L|}(-1)^{|S'|+|S''|}\right)\right]\cdot u^{\text{OR}}_{N\setminus L} \\
&\quad - \sum_{L\cap A=\emptyset,L\neq N\setminus A}\left[\sum_{S'\subseteq N\setminus A\setminus L}\left(\sum_{|S''|=0}^{|A|}C^{|S''|}_{|A|}(-1)^{|S'|+|S''|}\right)\right]\cdot u^{\text{OR}}_{N\setminus L} \qquad (15) \\
&= -(-1)\cdot u^{\text{OR}}_A - u^{\text{OR}}_\emptyset - \sum_{L\cap A\neq\emptyset,L\neq N}\left[\sum_{S'\subseteq N\setminus A\setminus L}0\right]\cdot u^{\text{OR}}_{N\setminus L} \\
&\quad - \sum_{L\cap A=\emptyset,L\neq N\setminus A}\left[\sum_{S'\subseteq N\setminus A\setminus L}0\right]\cdot u^{\text{OR}}_{N\setminus L} \\
&= u^{\text{OR}}_A - u^{\text{OR}}_\emptyset \\
&= u^{\text{OR}}_A.
\end{aligned}
$$

(3) Universal matching property of AND-OR interactions.

With the Universal matching property of AND interactions and the Universal matching property of OR interactions, we can easily get $v(\boldsymbol{x}_A) = u^{\text{AND}}_A + u^{\text{OR}}_A = v(\boldsymbol{x}_\emptyset) + \sum_{\emptyset\neq S\subseteq A} I^{\text{AND}}_S + \sum_{S:S\cap A\neq\emptyset} I^{\text{OR}}_S$, thus, we obtain the Universal matching property of AND-OR interactions.

■

Appendix C. Introductions of the Three Conditions for the Sparsity of AND Interactions

The sparsity of AND interactions has been proved by Ren et al. (2024) under the following three conditions.

Condition 1: The DNN does not encode AND interactions higher than the M-th order. That is, $\forall S \in \{S \subseteq N || S| \geq M+1\}, I_S^{\text{AND}} = 0$.

Condition 2: The average network output over all samples with m unmasked input variables increases when m becomes larger. That is, $\forall m' \leq m,\ \overline{v}^{(m')} \leq \overline{v}^{(m)}$, where $\overline{v}^{(m)} \triangleq \mathbb{E}_{|T|=m}[v(\boldsymbol{x}_T)]$.

Condition 3: When we decrease the number of unmasked input variables, *i.e.*, m, the decreasing speed of the average network output $\overline{v}^{(m)}$ is not faster than a polynomial function. That is, $\forall m' \leq m,\ \overline{v}^{(m')} \geq (\frac{m'}{m})^p\ \overline{v}^{(m)}$, where $p > 0$ is a positive constant.

Condition 1 can be derived by assuming no derivatives higher than the M-th order in the Taylor expansion of the function v, as proved by Ren et al. (2024). This is a common setting for a simple AI model or an AI model with almost zero high-order derivatives. Condition 2 means that a well-trained DNN is supposed to yield higher classification confidence for samples that are less masked. Condition 3 ensures that the classification confidence of the DNN does not significantly degrade on masked input samples. In real applications such as the image classification, the classification/detection of occluded samples is common. Thus, for a well-trained DNN, its confidence when classifying occluded (masked) samples should not be substantially lower than its confidence for unmasked samples.

Appendix D. Difficulty in Examining Whether an LLM Satisfies the Three Conditions for the Sparsity of AND Interactions

It is difficult for us to verify whether an LLM strictly satisfies the three conditions for the proof of Ren et al. (2024), due to the following three reasons. (1) The proof implicitly assumed that the DNN has a large number of input variables, *i.e.*, n was supposed to be large. For example, the DNN for image classification with thousands of pixels. However, the LLM for text generation does not have such a large number of input words. (2) The proof assumed that the occlusion/masking of input variables was common in training samples. However, except for masking-based subsequence learning, the masking of words is not a necessary setting for learning an LLM. (3) The LLM has many more parameters than traditional DNNs.

Appendix E. Discussions on Theorem 1

Because the output of an LLM can be represented as the sum of all 2^{n+1} AND-OR interactions, according to Theorem 1, we can explain the inference logic of a DNN by analyzing all 2^{n+1} AND-OR interactions. Furthermore, if we verify that among all 2^{n+1} AND-OR interactions, only a small number of interactions have salient effects on an LLM's inference scores, and the effects of the remaining interactions are almost zero, then we can use a small number of salient interactions to explain the LLM's inference logic. These few interactions can be considered symbolic patterns encoded by an LLM. Thus, we could explain the inference logic of an LLM in a symbolic manner.

Appendix F. Experimental Details and More Results

F.1 Details About How to Mask Input Words for Different LLMs

In this paper, we used interactions to analyze six LLMs for text generation, including the OPT-1.3B model, the OPT-13B model, the OPT-30B model, the GPT-J-6B model, the GPT-Neo-2.7B model, and the LLaMA-7B model. For the three OPT models, we use the "pad" token with the token $id = 1$ to mask the words in $N \setminus T$. For the GPT-J-6B model, we use the "extratoken_1" token with the token $id = 50257$ to mask the words in $N \setminus T$. For the GPT-Neo-2.7B model, we use the "endoftext" token with the token $id = 50256$ to mask the words in $N \setminus T$. For the LLaMA-7B model, there is no token without special semantics in the dictionary. Therefore, following Ren et al. (2023b), we use a special embedding vector $\boldsymbol{b}^*$ to mask the embedding of the first layer corresponding to the tokens decoded from the words in $N \setminus T$.

F.2 Details About How to Select Salient AND-OR Interactions

Ren et al. (2023a) proposed a method that iteratively removed interactions from all 2^n AND interactions, thus preserving K most salient interactions. We extend the method of Ren et al. (2023a) to select K most salient AND-OR interactions from all 2^{n+1} AND-OR interactions. Specifically, in each iteration, we remove an interaction from the remaining interactions, causing the minimal decrease in the value of $\rho^{\text{unfaith}} = \mathbb{E}_{T\subseteq N}[v(\boldsymbol{x}_T) - (\sum_{S\subseteq T} I_S^{\text{AND}} + \sum_{S\cap T\neq\emptyset} I_S^{\text{OR}})]^2$. This process continues until K interactions are left, at which point the iteration stops.

F.3 Details About Network Architectures and Chosen Layers for Experiments in Section 3.4

OPT-1.3B model. The OPT-1.3B model is composed of the following parts: one word embedding layer (namely Embedding Layer), one position embedding layer (namely OPT LearnedPositionalEmbedding), 24 OPT decoder modules (namely OPTDecoderLayer), and one linear output layer (namely Linear Layer). The architecture can be summarized as Embedding Layer→OPTLearnedPositionalEmbedding→[OPTDecoderLayer]*24→Linear Layer. Each module of OPTDecoderLayer contains a self-attention mechanism layer (namely OPTAttention), an activation function (namely ReLU), a layer normalization operation (namely LayerNorm), two fully connected layers (namely Linear), and a final layer normalization operation (namely LayerNorm). In our experiments in Section 3.4, we uniformly selected 6 out of the 24 OPTDecoderLayer modules and conducted experiments based on the output features of their LayerNorm layers. Specifically, the selected modules are named OPTDecoderLayer 3, OPTDecoderLayer 7, OPTDecoderLayer 11, OPTDecoderLayer 15, OPTDecoderLayer 19, and OPTDecoderLayer 23.

GPT-J-6B model. The GPT-J-6B model includes a single word embedding layer (namely Embedding Layer), followed sequentially by 28 Transformer blocks (namely GPTJBlock), culminating in an output layer normalization (namely LayerNorm) and a linear output layer (namely Linear Layer). This architecture can be succinctly described as: Embedding Layer→ [GPTJBlock]*28 → LayerNorm → Linear Layer. Delving into the details of each module of GPTJBlock, it comprises three integral components, including a layer

normalization (namely LayerNorm), a self-attention mechanism (namely GPTJAttention), and a feed-forward network (namely GPTMLP). In our experiments in Section 3.4, we uniformly select 7 out of the 28 GPTBlocks and conduct experiments based on the output features of the LayerNorm operation. Specifically, the selected modules are named GPTJBlock 3, GPTJBlock 7, GPTJBlock 11, GPTJBlock 15, GPTJBlock 19, GPTJBlock 23, and GPTJBlock 27.

LLaMA-7B model. The LLaMA-7B model is composed of various elements: a single word embedding layer (namely Embedding Layer), followed by a sequence of 32 Llama decoder layers (namely LlamaDecoderLayer), an output layer normalization layer (namely LlamaRMSNorm), and culminating in a linear output layer (namely Linear Layer). This architecture can be summarized as Embedding Layer $\rightarrow$ [LlamaDecoderLayer]*32 $\rightarrow$ LlamaRMSNorm $\rightarrow$ Linear Layer. Delving into each module of LlamaDecoderLayer, it integrates multiple components, including a self-attention mechanism layer (namely LlamaAttention), a feed-forward network layer (namely LlamaMLP) encompassing several linear layers and activation functions, an input layer normalization (namely Input LayerNorm), and a post-attention layer normalization (namely Post Attention LayerNorm), the latter serving to normalize the output from the self-attention mechanism layer. In our experiments in Section 3.4, we uniformly select 7 out of the 32 modules of LlamaDecoderLayer and conduct experiments based on the output features of the Post Attention LayerNorm operation of each selected module. Specifically, the selected modules are named LlamaDecoderLayer 1, LlamaDecoderLayer 6, LlamaDecoderLayer 11, LlamaDecoderLayer 16, LlamaDecoderLayer 21, LlamaDecoderLayer 26, and LlamaDecoderLayer 31.

F.4 Details About Experimental Data

F.4.1 Testing Sets for Examining the Stability of Interactions

To examine the stability of interactions over different samples, we collect different sets of sentences generated by the LLM. We must carefully select these sentences to ensure that different sentences have similar contents and the same target word y^{truth}. For example, given two sentences $\boldsymbol{x}^{(1)}$ = "*Diabetes is a disease that affects the body's ability to use*" and $\boldsymbol{x}^{(2)}$ = "*Diabetes is a condition that impairs the body's ability to use*," we hope to check whether we can extract similar interactions from the two sentences, when the LLM generates the token $y^{\text{truth}} = [glucoses]$. To this end, we revise samples in the SQuAD, the WikiQA, and the ARC data sets, respectively, to obtain about seventy sets of sentences. Each set contains two sentences. One comes from the above data set, denoted by $\boldsymbol{x}^{(2)}$, and another is obtained by abbreviating $\boldsymbol{x}^{(2)}$, denoted by $\boldsymbol{x}^{(1)}$. Table 1 shows some examples of testing sets for examining the stability of interactions across different samples.

Besides, to examine the consistency of interactions encoded by two LLMs (Φ_1, Φ_2), we collect a set of sentences $\{\boldsymbol{x}\}$. Given each sentence in the collected set, the two LLMs (Φ_1, Φ_2) generate the same token y^{truth}. We select samples in the SQuAD, the WikiQA, and the ARC data sets, respectively, to obtain about fifty-five sets of sentences. Given the six LLMs introduced in Section 3.2, there are 15 pairs of LLMs in total. Table 2 shows some examples for examining the consistency of interactions across different LLMs.

$\boldsymbol{x}^{(2)}$ = Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion y^{truth} = San	$\boldsymbol{x}^{(1)}$ = Super Bowl 50 was an American football game to determine the champion of (NFL) . (AFC) champion Denver Broncos defeated (NFC) champion y^{truth} = San
$\boldsymbol{x}^{(2)}$ = One of the most famous people born in Warsaw was Maria Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the y^{truth} = Nobel	$\boldsymbol{x}^{(1)}$ = A famous people born in Warsaw was Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the y^{truth} = Nobel
$\boldsymbol{x}^{(2)}$ = Huguenot numbers peaked near an estimated two million by 1562, concentrated mainly in the southern and central parts of France, about one-eighth the number of French Catholics. As Huguenots gained influence and more openly displayed their y^{truth} = faith	$\boldsymbol{x}^{(1)}$ = Huguenot numbers peaked near two million by 1562, concentrated in the southern and central parts of France, one-eighth of French Catholics. As Huguenots gained influence and more openly displayed their y^{truth} = faith

Table 1: Examples for examining the stability of interactions over different samples.

$\boldsymbol{x}$ = One of the most famous people born in Warsaw was Maria Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the y^{truth} = Nobel
$\boldsymbol{x}$ = The Apollo program, also known as Project Apollo, was the third United States human spaceflight program carried out by the National Aeronautics and Space Administration (NASA), which accomplished landing the first humans on the Moon from 1969 to y^{truth} = 1972
$\boldsymbol{x}$ = The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva Amazónica, Amazonía or usually Amazonia; French: Forêt amazonienne; Dutch: Amazoneregenwoud), also known in English as Amazonia or the Amazon Jungle, is a moist broadleaf forest that covers most of the Amazon basin of South y^{truth} = America
$\boldsymbol{x}$ = Starting in the late 1950s, American computer scientist Paul Baran developed the concept Distributed Adaptive Message Block Switching with the goal to provide a fault-tolerant, efficient routing method for telecommunication messages as part of a research program at the RAND Corporation, funded by the US Department of y^{truth} = Defense
$\boldsymbol{x}$ = The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and y^{truth} = Norway

$\boldsymbol{x}$ = The immune system is a system of many biological structures and processes within an organism that protects against disease. To function properly, an immune system must detect a wide variety of agents, known as pathogens, from viruses to parasitic worms, and distinguish them from the organism's own
y^{truth} = cells

Table 2: Examples for examining the consistency of interactions over the OPT-1.3B model and the OPT-30B model.

F.4.2 Testing Sets for Checking Coarse-to-Fine Inference Patterns

We ask ChatGPT to provide us with some testing sentences that require complex expertise. Specifically, we give ChatGPT the following prompt, "Please tell me 10 theories from the field of [x], with each theory introduced in one sentence," where [x] can be physics or economics in this experiment. Then, ChatGPT generates some sentences based on our prompts, such as "The Big Bang theory, along with related cosmological research, reveals the origin, evolution, and structure of the universe." For this sentence, we input "The Big Bang theory, along with related cosmological research, reveals the origin, evolution, and structure of the" into the LLM used for our experiments (such as the OPT-1.3B model). If we observe that the output of the LLM is indeed "universe," then we retain this sentence for the experiment.

F.5 Details About How to Select Domain-Specific Tokens and Salient Interactions in Section 3.3

Let Y^{all} denote the set of all tokens encoded by an LLM. Given an input sentence $\boldsymbol{x}$, each token $y \in Y^{\text{all}}$ has an output score $z(y) \in \mathbb{R}$. We identify 20 tokens with the top 20 highest scores $z(y)$, and let Y^{all} contain only these 20 tokens for simplicity. For example, let the OPT-1.3B model be given the input sentence $\boldsymbol{x}$ = "The Big Bang theory, along with related cosmological research, reveals the origin, evolution, and structure of the," the selected 20 tokens with the top 20 highest scores $z(y)$ are Y^{all} = {*universe, Universe, cosmos, observable, entire, Big, cosmic, early, known, big, matter, material, first, galaxies, world, solar, Milky, physical, fundamental, large*}. Then, we select tokens relevant to the domain of physics from Y^{all}, and obtain the domain-specific set Y^{related} = {*universe, Universe, cosmos, observable, cosmic, matter, material, galaxies, world, solar, physical, fundamental*}.

In terms of selecting salient interactions, for each sample $\boldsymbol{x}$, we select interactions $\{S \mid |I_S^{\text{AND}}| > 0.03 \cdot \mathbb{E}_{\boldsymbol{x}'}|v(\boldsymbol{x}') - v(\boldsymbol{x}_\emptyset)|\}$ and $\{S \mid |I_S^{\text{OR}}| > 0.03 \cdot \mathbb{E}_{\boldsymbol{x}'}|v(\boldsymbol{x}') - v(\boldsymbol{x}_\emptyset)|\}$ as salient interactions.

F.6 Settings for the Fair Comparison of Sparsity Between LLMs

We also consider the fair comparison of sparsity between LLMs. Different LLMs have different sizes of dictionaries. The dictionaries of all GPT models contain 50,257 tokens. The dictionaries of all OPT models contain 50,272 tokens. The dictionary of the LLaMA model contains 32,000 tokens. To normalize the variation of dictionary size over different LLMs, we put the output dimension for y^{truth} and the other 999 output dimensions into a

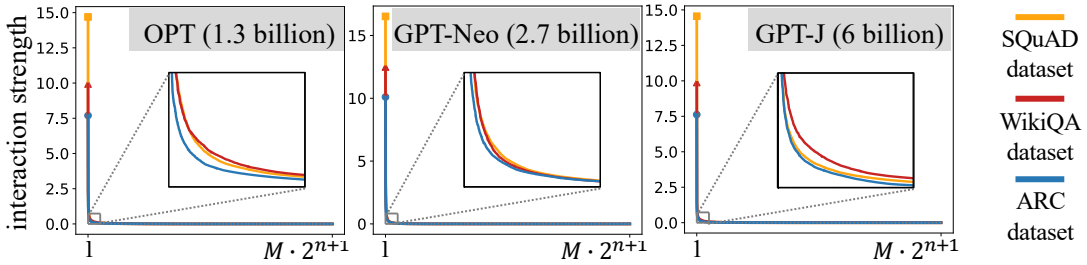


Figure 23: Sparsity of AND-OR interactions extracted from different input samples. We show the absolute values of AND-OR interactions in a descending order.

softmax layer to compute the probability $p(y = y^{\text{truth}}|\boldsymbol{x})$, where the 999 output dimensions (corresponding to 999 tokens) are uniformly selected from all output dimensions.

F.7 More Experimental Results

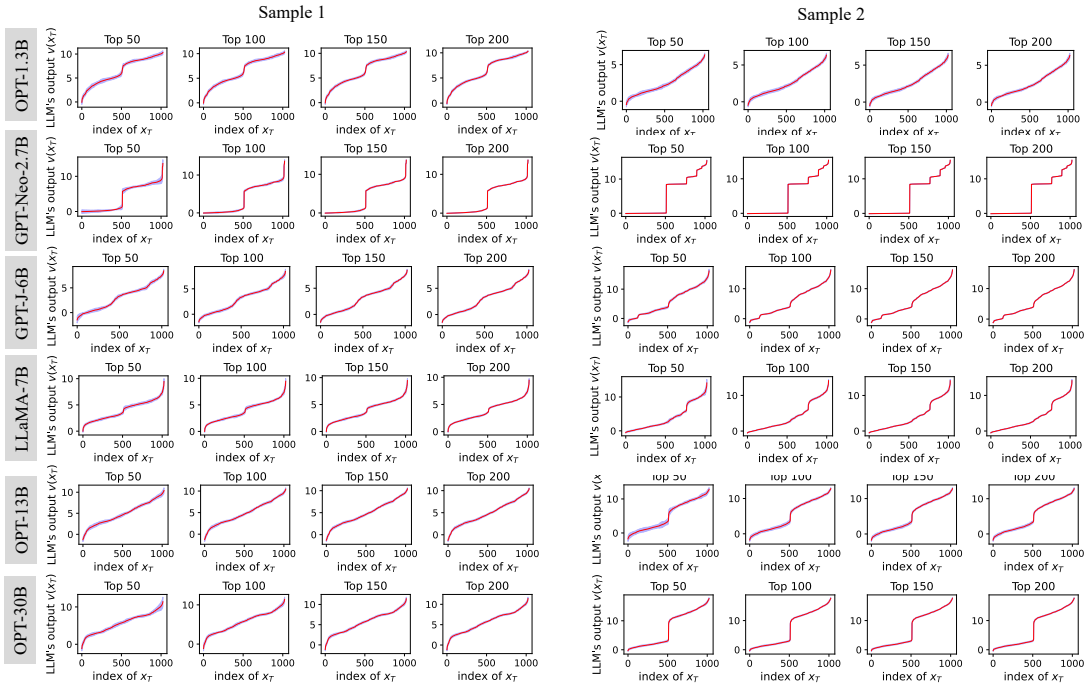


Figure 24: Quality of universal matching. The red line shows model outputs on all 2^n masked samples in an ascending order, given a specific input sentence. We use the matching error averaged over neighboring 50 subsets T for a smoothed visualization of the matching error, *i.e.*, $error_T$.

Fig. 23 shows that an LLM's output scores can be disentangled into a small number of interactions. Fig. 24 shows that we can use a small number of salient AND-OR interactions to well estimate the LLM's outputs on different masked sentences. Note that both in Fig. 3(b) and Fig. 24, we use the matching error averaged over neighboring 50 subsets T for

a smoothed visualization of the matching error, *i.e.*, $error_T$. Fig. 25 shows more examples of the LLM providing random responses, which also indicate that the extraction of high-order and offsetting interactions is a common phenomenon when the LLM fails to extract meaningful information for inference.

(a) **Input:** Question: The wood[a] crafter complains that coins[a] are typically minted from metals rather than wood[b]. Which material are coins[b] made of? Answer: The coins are made of
The GPT-Neo-2.7B model **predict:** 9).
(the following interactions are computed using the first output token "9") $v(\text{"9"}) = -0.23$

{coins[a], minted, metals, rather, wood[b], coins[b], made}	$I_{\text{and}}(S) = 0.13$
{made}	$I_{\text{and}}(S) = -0.13$
{crafter}	$I_{\text{and}}(S) = 0.11$
{crafter, coins[a], minted, metals, rather, wood[b]}	$I_{\text{and}}(S) = -0.11$
{coins[a], minted}	$I_{\text{or}}(S) = -0.11$
{crafter, coins[a], minted, metals, rather, wood[b], coins[b]}	$I_{\text{and}}(S) = -0.10$
{minted, metals, material, coins[b], made}	$I_{\text{and}}(S) = 0.10$
{crafter, coins[a], minted, metals, rather, wood[b], material, coins[b], made}	$I_{\text{and}}(S) = -0.10$
{coins[a], material, coins[b], made}	$I_{\text{and}}(S) = 0.09$
{crafter, coins[a], minted, coins[b], made}	$I_{\text{and}}(S) = 0.09$

(b) **Input:** Emma[a] is the daughter of Mary. Mary's favorite dish is pasta. Emma[b] comes from
The LLaMA-7B model **predict:** 2 parents.
(the following interactions are computed using the first output token "2") $v(\text{"2"}) = 0.79$

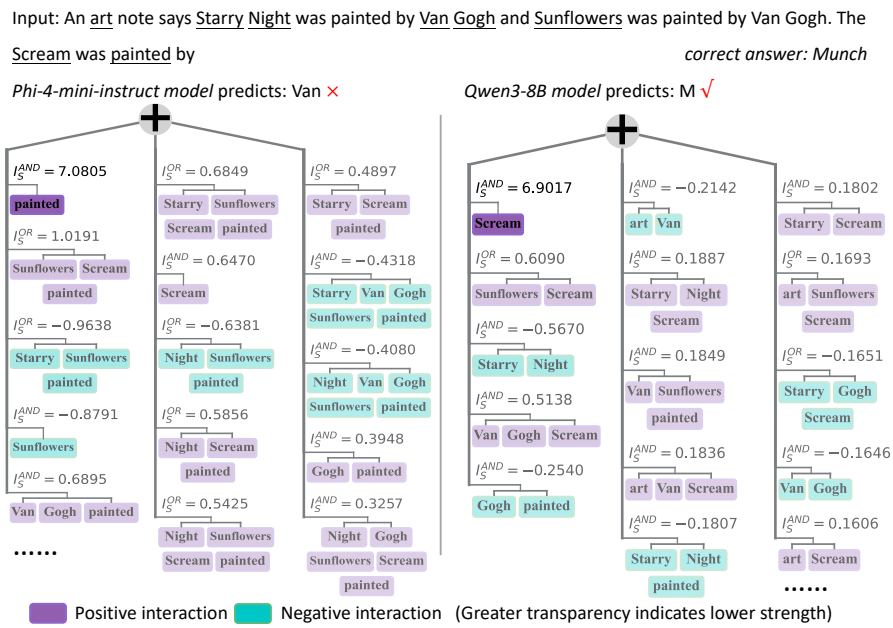
{comes, from}	$I_{\text{or}}(S) = 0.86$
{comes, from}	$I_{\text{and}}(S) = -0.85$
{Emma[a], Emma[b], comes, from}	$I_{\text{and}}(S) = -0.57$
{Emma[a]}	$I_{\text{and}}(S) = -0.52$
{favorite, dish, pasta, Emma[b], comes, from}	$I_{\text{and}}(S) = -0.51$
{from}	$I_{\text{and}}(S) = 0.47$
{Mary's, favorite, dish, Emma[b], comes, from}	$I_{\text{and}}(S) = -0.47$
{Mary, Mary's, favorite, dish, pasta, comes, from}	$I_{\text{and}}(S) = -0.46$
{daughter, Mary, Mary's, favorite, dish, pasta, Emma[b], from}	$I_{\text{and}}(S) = -0.46$
{Emma[a], Mary's, favorite, Emma[b], comes, from}	$I_{\text{and}}(S) = 0.43$

(c) **Input:** Kevin loves playing the guitar and writing songs. His favorite type of weather is
The LLaMA-7B model **predict:** 70 degrees.
(the following interactions are computed using the first output token "7") $v(\text{"7"}) = 2.53$

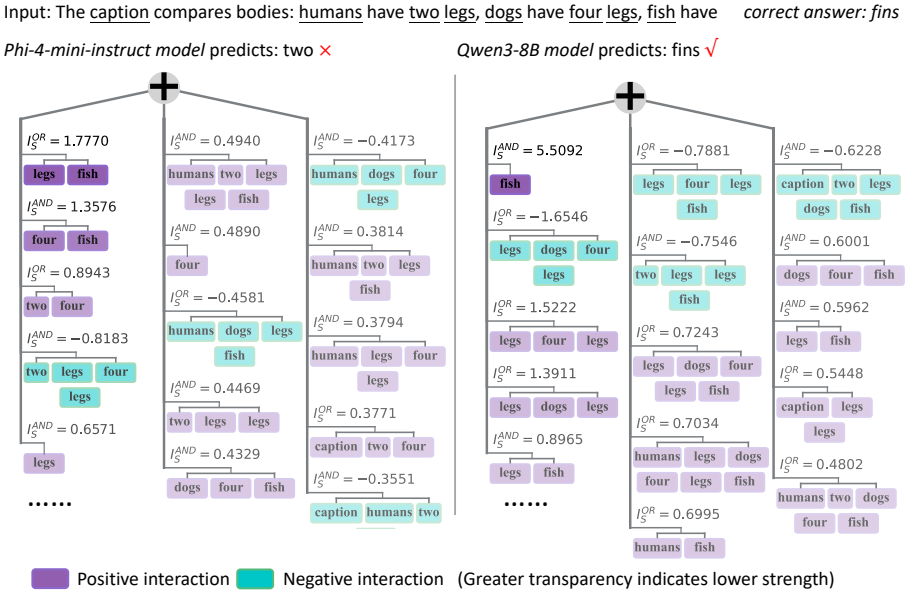
{His, favorite, type}	$I_{\text{or}}(S) = 1.27$
{loves, playing, guitar, writing, songs, His, favorite, type, weather}	$I_{\text{and}}(S) = -1.25$
{His, favorite, type, weather}	$I_{\text{or}}(S) = -1.20$
{favorite, type}	$I_{\text{or}}(S) = -1.13$
{His, type}	$I_{\text{or}}(S) = -1.07$
{His}	$I_{\text{and}}(S) = 0.90$
{His, favorite}	$I_{\text{or}}(S) = -0.82$
{loves, playing, His, favorite, type, weather}	$I_{\text{and}}(S) = 0.80$
{Kevin, His, favorite, type, weather}	$I_{\text{and}}(S) = 0.77$
{favorite}	$I_{\text{and}}(S) = 0.73$
{weather}	$I_{\text{and}}(S) = 0.71$
{loves, playing, writing, songs, favorite, type, weather}	$I_{\text{and}}(S) = 0.69$
{loves, writing, His, favorite, type, weather}	$I_{\text{and}}(S) = 0.63$
{Kevin, loves, playing, His, favorite, type, weather}	$I_{\text{and}}(S) = -0.61$
{Kevin, loves, playing, guitar, writing, songs, His, favorite, weather}	$I_{\text{and}}(S) = 0.60$
{Kevin, loves, playing, writing, songs, favorite, type, weather}	$I_{\text{and}}(S) = -0.57$

Figure 25: More examples of the LLM providing random responses.

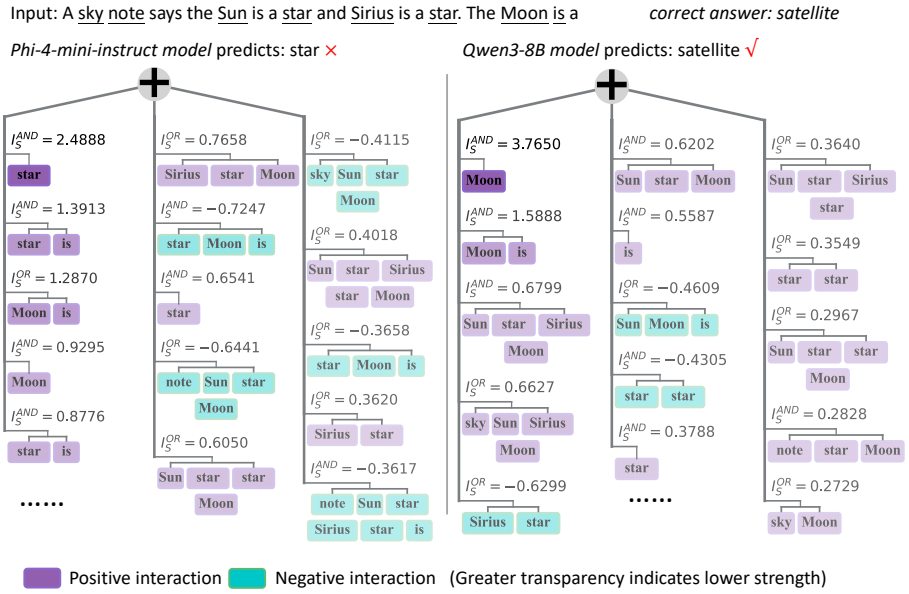
F.8 Additional Hallucination Cases



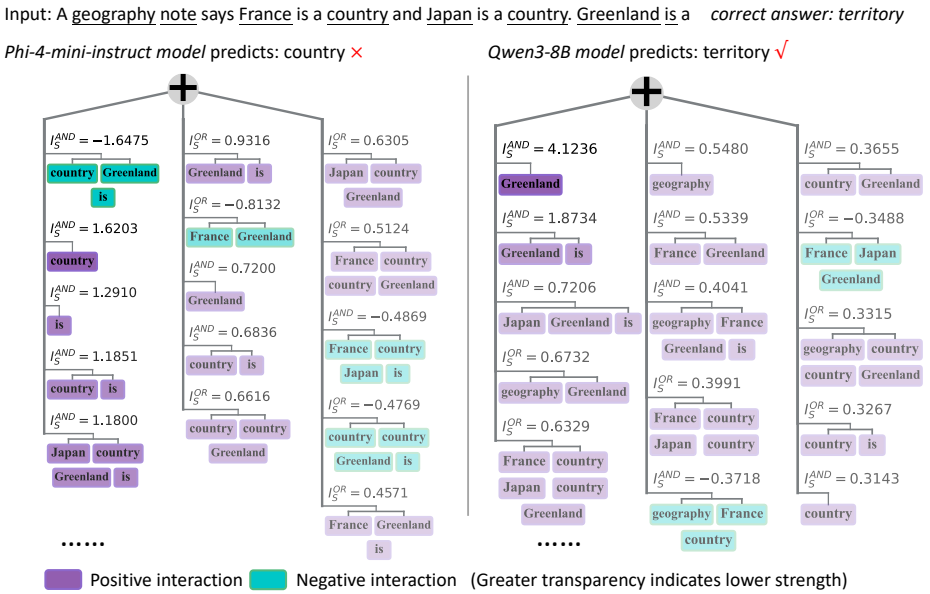
(a) Phi-4-mini-instruct focuses on irrelevant interactions in the previous context, such as "*Sunflowers, Scream, painted*" and "*Van Gogh, painted*," transferring the painter pattern of *Starry Night* and *Sunflowers* to *The Scream*. In comparison, Qwen3-8B better identifies "*Scream*," assigning this token a much higher effect ($I_S^{\text{AND}} = 6.9017$), explaining why it gives the correct result.



(b) Phi-4-mini-instruct focuses on irrelevant interactions in the previous context, such as "*legs, fish*," "*four, fish*," and "*two, four*," transferring leg-count facts about humans and dogs to fish. In comparison, Qwen3-8B better identifies "*fish*," assigning this token a much higher effect ($I_S^{\text{AND}} = 5.5092$), explaining why it predicts "*fins*" correctly.



(c) Phi-4-mini-instruct focuses on irrelevant interactions in the previous context, such as "*star*," "*star, is*," and "*Moon, is*," transferring the star category of the Sun and Sirius to the Moon. In comparison, Qwen3-8B better identifies "*Moon*," assigning this token a much higher effect ($I_S^{\text{AND}} = 3.7650$), explaining why it predicts "*satellite*" correctly.



(d) Phi-4-mini-instruct focuses on irrelevant interactions in the previous context, such as "*country*," "*country, is*," and "*Japan, country, Greenland, is*," transferring the country category of France and Japan to Greenland. In comparison, Qwen3-8B better identifies "*Greenland*," assigning this token a much higher effect ($I_S^{\text{AND}} = 4.1236$), explaining why it predicts "*territory*" correctly.

Figure 26: More examples of the LLM providing hallucination responses.

References

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for Shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019.

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

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL `https://doi.org/10.5281/zenodo.5297715`. If you use this software, please cite it using these metadata.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. InternLM2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

Lu Chen, Siyu Lou, Benhao Huang, and Quanshi Zhang. Defining and extracting generalizable interaction primitives from DNNs. In *The Twelfth International Conference on Learning Representations*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Ian Covert, Scott Lundberg, and Su-In Lee. Feature removal is a unifying principle for model explanation methods. *arXiv preprint arXiv:2011.03623*, 2020.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models implicitly perform gradient descent as meta-optimizers, 2022.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. Discovering and explaining the representation bottleneck of DNNs. In *International Conference on Learning Representations*, 2022.

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, Ziwei Yang, Zheyang Li, and Quanshi Zhang. Unifying fourteen post-hoc attribution methods with taylor interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2950–2958, 2019.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL `https://arxiv.org/abs/2407.21783`.

Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384, 2006.

Yuling Gu, Bhavana Dalvi Mishra, and Peter Clark. Do language models have coherent mental models of everyday things?, 2022.

Hannes Hansen and Martin N. Hebart. Semantic features of object concepts generated with GPT-3, 2022.

Bairu Hou, Yang Zhang, Jacob Andreas, and Shiyu Chang. A probabilistic framework for LLM hallucination detection via belief tree propagation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3076–3099, 2025.

Fangbo Li, Junpeng Zhang, Qihan Ren, and Quanshi Zhang. How does parameter pruning reshape DNN representations? an interaction-driven exploration. *arXiv preprint arXiv:2609.06483*, 2026.

Mingjie Li and Quanshi Zhang. Does a neural network really encode symbolic concepts? *Proceedings of the 40th International Conference on Machine Learning*, 2023a.

Mingjie Li and Quanshi Zhang. Technical note: Defining and quantifying AND-OR interactions for faithful and concise explanation of DNNs. *arXiv preprint arXiv:2304.13312*, 2023b.

Siyu Lou, Yao Yan, Yuntian Chen, and Quanshi Zhang. Cross-LLM consistency in inference: Evidence from shared interactions. *arXiv preprint arXiv:2606.08129*, 2026. URL `https://arxiv.org/abs/2606.08129`.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

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

Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi-ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025. URL `https://arxiv.org/abs/2503.01743`.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, et al. Qwen2.5 technical report, 2024. URL `https://arxiv.org/abs/2412.15115`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, art. arXiv:1606.05250, 2016.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in DNNs. *IEEE Conference on Computer Vision and Pattern Recognition*, 2023a.

Jie Ren, Zhanpeng Zhou, Qirui Chen, and Quanshi Zhang. Can we faithfully represent masked states to compute Shapley values on a DNN? In *International Conference on Learning Representations*, 2023b.

Qihan Ren, Huiqi Deng, Yunuo Chen, Siyu Lou, and Quanshi Zhang. Bayesian neural networks tend to ignore complex and sensitive concepts. *arXiv preprint arXiv:2302.13095*, 2023c.

Qihan Ren, Jiayang Gao, Wen Shen, and Quanshi Zhang. Where we have arrived in proving the emergence of sparse interaction primitives in DNNs. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, 2024.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Pritish Sahu, Michael Cogswell, Yunye Gong, and Ajay Divakaran. Unpacking large language models with conceptual consistency, 2022.

Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024.

Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, et al. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. *arXiv preprint arXiv:2008.05122*, 2020.

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. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Xin Wang, Shuyun Lin, Hao Zhang, Yufei Zhu, and Quanshi Zhang. Interpreting attributions and interactions of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1095–1104, 2021a.

Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. In *International Conference on Learning Representations*, 2021b.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do Llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, et al. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, 2015.

Yuchen Yang, Houqiang Li, Yanfeng Wang, and Yu Wang. Improving the reliability of large language models by leveraging uncertainty-aware in-context learning. *arXiv preprint arXiv:2310.04782*, 2023.

Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Extraction of an explanatory graph to interpret a CNN. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3863–3877, 2020.

Quanshi Zhang, Xin Wang, Jie Ren, Xu Cheng, Shuyun Lin, Yisen Wang, and Xiangming Zhu. Proving common mechanisms shared by twelve methods of boosting adversarial transferability. *arXiv preprint arXiv:2207.11694*, 2022a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022b.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. Enhancing uncertainty-based hallucination detection with stronger focus. *arXiv preprint arXiv:2311.13230*, 2023.

Huilin Zhou, Huijie Tang, Mingjie Li, Hao Zhang, Zhenyu Liu, and Quanshi Zhang. Explaining how a neural network play the go game and let people learn. *arXiv preprint arXiv:2310.09838*, 2023a.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Concept-level explanation for the generalization of a DNN. *arXiv preprint arXiv:2302.13091*, 2023b.