# RECONCILING CONTRADICTORY VIEWS ON THE EFFECTIVENESS OF SFT IN LLMS: AN INTERACTION PERSPECTIVE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

This paper explores a scientific question in supervised fine-tuning (SFT): why SFT is broadly effective for small-scale deep neural networks, yet can produce inconsistent or even detrimental effects when applied to large language models (LLMs). Recent advances in interaction-based explanations (Ren et al., 2024a) suggest that interactions between words/tokens provide a faithful metric for quantifying the inference patterns encoded by LLMs. We find that the evolution of interactions during SFT can effectively explain the inconsistent effectiveness of SFT for LLMs. Specifically, we find that (1) SFT primarily removes noise-like interactions, while rarely acquiring reliable new interactions. (2) This denoising stage is extremely brief, after which continued fine-tuning tends to introduce overfitted interactions. We validate these findings across multiple LLMs and datasets. Our findings provide new insights into early stopping and offer practical guidance for LLM training.

## 1 INTRODUCTION

Supervised fine-tuning (SFT) has become a common strategy for adapting pretrained deep neural networks (DNNs) to domain-specific applications. However, its effectiveness in large language models (LLMs) has been subject to inconsistent and even contradictory viewpoints. While some prior studies (Ouyang et al., 2022; Chung et al., 2024; Wei et al., 2021) considered SFT essential for improving instruction-following ability and overall usability, others (Luo et al., 2025; Wang et al., 2022; Shi et al., 2024; Gudibande et al., 2023) suggested that it may overfit to the fine-tuning data, limiting generalization and even degrading performance.

Therefore, rather than drawing a premature conclusion about the utility of SFT or proposing a new SFT algorithm, this paper explores the following scientific question: **what internal factors account for the inconsistent effectiveness of SFT across different LLMs?** Specifically, we investigate whether verifiable metrics can quantify the inference patterns learned by LLMs during SFT, with these patterns explaining the disparate generalization performance across samples during SFT.

**Background.** As an emerging explanation strategy, Ren et al. (2024a) measure the interactions between input variables encoded by a DNN as primitive inference patterns. For example, the interactions in an LLM usually represent phrase patterns between different words/tokens.

> As illustrated in Figure 1, given a physics-related input prompt, an LLM may automatically learn a phrase pattern between the words *"laws," "of,"* and *"motion,"* which can be quantified as an interaction. When all three words are present in the input prompt, this interaction is activated and contributes an effect of $0.41$ to boost the confidence in generating the token *"acceleration."* More importantly, a substantial body of empirical studies (Li & Zhang, 2023b; Ren et al., 2023a; 2024a; Chen et al., 2024; Zhou et al., 2024) and theoretical theorems[1]have rigorously demonstrated the faithfulness of using interactions to explain the inference patterns in LLMs.

**Our work.** In this study, we find that the evolution of the representation quality of interactions can well explain the inconsistent effectiveness of SFT across different LLMs. Specifically, as illustrated in Figure 1, we explore how new interactions emerge and how existing interactions are removed throughout the SFT process, yielding several key findings.

[1]Please see Section 3.1 and Appendix A for details.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

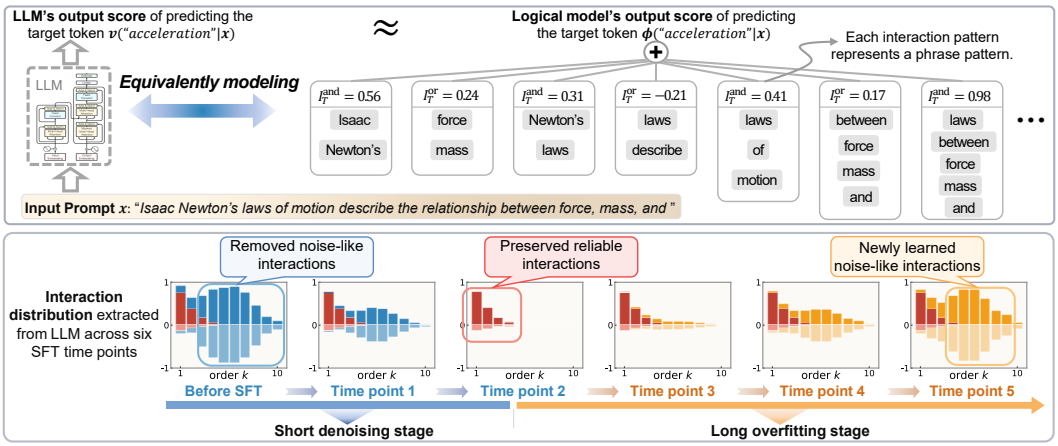


Figure 1: (Top) The complex inference patterns of an LLM can be faithfully represented by a few interactions (phrase patterns) in a logical model $\phi(\cdot)$. (Bottom) The SFT first enters a short denoising stage of removing noise-like interactions, which are non-generalizable and mutually canceling. In the subsequent prolonged overfitting stage, the LLM gradually learns new noise-like interactions again.

**(1) LLMs primarily remove noise interactions during SFT while rarely learning reliable ones.** We explore how the interactions encoded by an LLM evolve throughout the SFT process, and categorize the changed interactions into three types: *removed interactions*, *newly emerged interactions*, and *preserved interactions*.

*Removed interactions* refer to interactions that are eliminated during SFT. We find that most removed interactions represent noise patterns, for three reasons. First, their positive and negative effects largely cancel out, *i.e.,* approximately half of the removed interactions contribute positively to the prediction of target tokens, while the other half contribute negatively. Second, these interactions tend to exhibit higher complexity, where the complexity (order) of an interaction is quantified as the number of input variables involved in the interaction. Third, they show weak generalizability, meaning they are less likely to generalize across different LLMs.

*Newly emerged interactions* are interactions that are newly acquired during SFT. Similar to removed interactions, most newly emerged interactions also correspond to non-generalizable noise patterns.

*Preserved interactions* usually represent reliable patterns. They tend to have low complexity and contribute positively to the prediction of target tokens, with limited cancellation between positive and negative effects. Moreover, they exhibit relatively strong generalization across different LLMs.

**(2) The removal of noise interactions occurs only at the very early stage of SFT,** *e.g.,* within the first 1000 training steps, whereas further fine-tuning primarily introduces overfitted interactions. Moreover, we find that the emergence of these overfitted patterns is strongly correlated with downstream task performance no longer improving and often declining.

In summary, our findings suggest that SFT is indeed effective, yet its effective regime is surprisingly short. A small number of training samples is sufficient to remove a substantial fraction of noise interactions. Beyond this regime, continued fine-tuning is likely to introduce overfitted interactions. These results suggest that interactions can serve as diagnostic signals for monitoring SFT and provide a more principled criterion for early stopping in end-to-end SFT.

## 2 Related Work

**SFT of LLMs.** SFT has been widely used to adapt pretrained models to downstream tasks (Devlin et al., 2019; Raffel et al., 2020). Prior studies (Ouyang et al., 2022; Wei et al., 2021; Chung et al., 2024; Zhou et al., 2023) show that SFT can improve zero-shot generalization, instruction following, and task-specific performance. However, its effectiveness remains debated. SFT may over-specialize models to task formats and weaken their general in-context learning ability (Wang et al., 2022); it may imitate the response style of stronger models without acquiring their underlying capabilities

(Gudibande et al., 2023); and continual instruction tuning may cause catastrophic forgetting of previously acquired knowledge and reasoning abilities (Luo et al., 2025).

**Interaction-based explanation.** Interaction-based explanation has recently attracted increasing attention for explaining DNNs, as surveyed in (Zhou et al., 2025). A large body of work studies how the complex inference logic of DNNs can be characterized by interactions (Li & Zhang, 2023b; Ren et al., 2023a; 2024a; Chen et al., 2024; Zhou et al., 2024). For example, Ren et al. (2024a) show that the output score of a DNN can be faithfully predicted by a small number of AND-OR interactions. Zhou et al. (2024) further use interactions to analyze the generalization power of DNNs, while Wang et al. (2020) use interactions to explain adversarial transferability.

In this study, we show that interactions provide an interpretable and verifiable metric for characterizing SFT in LLMs. SFT first performs rapid denoising within a brief stage, but subsequent training is dominated by overfitting. This suggests that SFT for LLMs is necessary, yet challenges the expectation that large-scale fine-tuning on massive datasets consistently yields further gains.

# 3 Understanding Representation Gain and Loss of LLM in SFT

## 3.1 Preliminaries: Interaction-Based Explanation

Given an LLM $v$ and an input prompt $\boldsymbol{x}$, which consists of $n$ input variables[2] indexed by the set $N = \{1, 2, ..., n\}$, the LLM's output score $v(\boldsymbol{x}) \in \mathbb{R}$ is usually (Chen et al., 2026; Ren et al., 2024b; Deng et al., 2021) defined as the following confidence score of generating a sequence of the target m tokens $[y_1, y_2, \cdots, y_m]$.

$$v(\boldsymbol{x}) \overset{\text{def}}{=} \sum_{i=1}^{m} \log \frac{p(y = y_i|\boldsymbol{x}, \boldsymbol{y}_i^{\text{preceding}})}{1 - p(y = y_i|\boldsymbol{x}, \boldsymbol{y}_i^{\text{preceding}})} \in \mathbb{R}, \tag{1}$$

where $\boldsymbol{y}_i^{\text{preceding}} \overset{\text{def}}{=} [y_1, y_2, \cdots, y_{i-1}]$ denotes the sequence of the preceding $(i-1)$ tokens before generating the $i$-th token. $p(y = y_i|\boldsymbol{x}, \boldsymbol{y}_i^{\text{preceding}})$ denotes the probability that the LLM predicts the target token $y_i$ at step $i$. Notably, $\boldsymbol{y}_1^{\text{preceding}} = []$.

Thus, the goal of interaction-based explanation is to disentangle a set of AND-OR interactions from the LLM as primitive inference patterns to compute the score $v(\boldsymbol{x})$. Specifically, Chen et al. (2024) construct the following logical model composed of AND-OR interactions, where each interaction represents a phrase pattern automatically used by the LLM $v$ on a given input $\boldsymbol{x}$.

$$\forall \boldsymbol{x}' \in \Psi,\ \phi(\boldsymbol{x}') \overset{\text{def}}{=} \sum_{T \in \Omega^{\text{and}}} \underbrace{I_T^{\text{and}} \cdot \mathbb{1}(\substack{\boldsymbol{x}' \text{ triggers AND relation} \\ \text{between input variables in } T})}_{\text{an AND interaction between input variables in } T} + \sum_{T \in \Omega^{\text{or}}} \underbrace{I_T^{\text{or}} \cdot \mathbb{1}(\substack{\boldsymbol{x}' \text{ triggers OR relation} \\ \text{between input variables in } T})}_{\text{an OR interaction between input variables in } T} + b, \tag{2}$$

where $b$ is a scalar bias. $\Omega^{\text{and}}$ and $\Omega^{\text{or}}$ represent the set of AND interactions and the set of OR interactions, respectively. $I_T^{\text{and}} \in \mathbb{R}$ and $I_T^{\text{or}} \in \mathbb{R}$ denote numerical effects of interactions.

The AND trigger function $\mathbb{1}(\substack{\boldsymbol{x}' \text{ triggers AND relation} \\ \text{between input variables in } T}) \in \{0, 1\}$ represents an AND relation between a set $T \subseteq N$ of input variables[2]. It returns 1 if all variables in $T$ are present (not masked[3]) in $\boldsymbol{x}'$. The OR trigger function $\mathbb{1}(\substack{\boldsymbol{x}' \text{ triggers OR relation} \\ \text{between input variables in } T}) \in \{0, 1\}$ represents an OR relation between a set $T \subseteq N$ of input variables[2]. It returns 1 whenever any variable in $T$ is present (not masked[3]) in $\boldsymbol{x}'$.

**Universal matching property.** Crucially, Chen et al. (2024) propose a method to compute interaction effects $I_T^{\text{and}}$ and $I_T^{\text{or}}$ by analyzing the LLM output $v(\cdot)$ (see Appendix F for details and pseudocode). Furthermore, they prove that **the logical model $\phi$, constructed based on the extracted AND-OR interactions, can accurately predict the LLM outputs over all $2^n$ masked[3] states** in $\Psi$.

$$\forall \boldsymbol{x}' \in \Psi,\ |\phi(\boldsymbol{x}') - v(\boldsymbol{x}')| < \epsilon, \tag{3}$$

where $\Psi = \{\boldsymbol{x}_S | S \subseteq N\}$ denotes $2^n$ masked states of $\boldsymbol{x}$. $\boldsymbol{x}_S$ denotes a masked[3] state that retains only the input variables in $S$, with all variables in $N \setminus S$ being masked[3]. $\epsilon$ is a small scalar to guarantee the fidelity of the explanations.

[2] An input variable can be defined as the embedding vector(s) corresponding to a token, word, or phrase. In this paper, we follow (Chen et al., 2024) to define each input variable as the embedding vector(s) of a word.

[3] An input word is masked by replacing the embedding vector(s) of its corresponding token(s) with a baseline vector $\boldsymbol{b} \in \mathbb{R}^d$. We follow (Chen et al., 2026) to set $\boldsymbol{b}$ to represent a *no-information* state of the input.

Please see Appendices D.4 and A for empirical results on explaining LLMs and theoretical analyses demonstrating the high fidelity of interaction-based explanations, respectively. *The supplementary material includes a **video demo** showing how interactions explain inference patterns in LLMs.*

**Sparsity of interactions.** Ren et al. (2024a) prove that the number of interactions extracted by a DNN from a given input sample is relatively small and theoretically bounded, typically ranging from 50 to 150 interactions in practice. Please see Appendix B for empirical validation of the sparsity.

**Computational cost & efficient solutions.** Similar to the computation of Shapley values (Shapley et al., 1953), extracting interaction effects incurs exponential computational cost. However, prior studies have proposed approximation algorithms (Kang et al., 2024; 2025; Butler et al., 2025), as well as practical engineering techniques (Li & Zhang, 2023a), to enable efficient interaction extraction. These advances facilitate practical applications of interaction-based explanations (see Appendix C).

### 3.2 Analyzing SFT of LLMs through Different Types of Interactions

Conducting SFT on a target dataset has been regarded as an effective approach for adapting pre-trained DNNs (Devlin et al., 2019; Raffel et al., 2020). However, a growing debate has emerged regarding whether SFT remains consistently effective for LLMs. Some studies (Luo et al., 2025; Wang et al., 2022; Shi et al., 2024; Gudibande et al., 2023) argued that directly applying SFT to LLMs yields limited performance gains and may instead introduce unpredictable risks of overfitting.

Therefore, in this study, **we aim to identify the internal factors that account for the inconsistent effectiveness of SFT across different LLMs.** To this end, interaction has been widely considered (Li & Zhang, 2023b; Ren et al., 2023a; 2024a; Chen et al., 2024; Zhou et al., 2024) as a reliable metric for analyzing the internal factors (inference patterns) of a DNN.

In this way, by continuously tracking interactions in LLMs throughout SFT, we uncover that (1) **the SFT primarily eliminates noise interactions while rarely acquiring reliable ones**, and (2) **this denoising stage is extremely brief, after which continued fine-tuning leads to overfitting**.

#### 3.2.1 Three Types: Newly emerged, Removed, and Preserved Interactions

To better characterize the improvement and degradation of representation quality in LLMs during SFT, we categorize interactions into three types: (1) removed interactions, which are initially encoded by the LLM but subsequently eliminated by the SFT; (2) preserved interactions, which are retained throughout the SFT; and (3) newly emerged interactions, which are acquired during SFT.

Let $\Omega_t^{\text{and}}$ and $\Omega_t^{\text{or}}$ denote the sets of AND and OR interactions encoded by the LLM after $t$ gradient-update steps of SFT, respectively. In particular, $t = 0$ denotes the time point before SFT. We then quantify the three types of interactions at each time point as follows.

• *Type 1*: **Removed interactions** at time point $t$ refer to interactions that are encoded by the LLM before SFT but are eliminated within the first $t$ optimization steps[4] of SFT. The set of removed interactions is identified as follows.

$$\forall \text{type} \in \{\text{and}, \text{or}\}, \quad \boldsymbol{R}_t^{\text{type}} \overset{\text{def}}{=} \Omega_0^{\text{type}} \setminus \Omega_t^{\text{type}}. \tag{4}$$

• *Type 2*: **Preserved interactions** at time point $t$ refer to interactions that are encoded by the LLM before SFT and remain encoded throughout the first $t$ optimization steps[4] of SFT. The set of preserved interactions is collected as follows.

$$\forall \text{type} \in \{\text{and}, \text{or}\}, \quad \boldsymbol{P}_t^{\text{type}} \overset{\text{def}}{=} \boldsymbol{P}_{t-1}^{\text{type}} \cap \Omega_t^{\text{type}}. \tag{5}$$

• *Type 3*: **Newly emerged interactions** at time point $t$ refer to interactions that are newly learned during the first $t$ optimization steps[4] of SFT. Newly emerged interactions are identified as the complement of the preserved interactions within all interactions encoded at time point $t$, as follows.

$$\forall \text{type} \in \{\text{and}, \text{or}\}, \quad \boldsymbol{E}_t^{\text{type}} \overset{\text{def}}{=} \Omega_t^{\text{type}} \setminus \boldsymbol{P}_t^{\text{type}}. \tag{6}$$

[4]An optimization step refers to one parameter-update operation during SFT, typically performed after computing gradients on one or more mini-batches of training samples.

To further assess the quality of newly emerged, removed, and preserved interactions during the SFT process, we introduce the following metrics to quantify the representational quality of interactions.

**Order/complexity of interactions.** The order (or complexity) of an interaction $T$ is defined as the number of input variables involved in $T$, *i.e.*, $\text{order}(T) \overset{\text{def}}{=} |T|$.

It has been found that low-order interactions usually represent more reliable inference patterns than high-order interactions (Ren et al., 2024b; Zhou et al., 2024; Ren et al., 2023b). (1) Low-order interactions are found to be more robust to input noise (Ren et al., 2023b). (2) Zhou et al. (2024) demonstrate that low-order interactions exhibit stronger generalizability than high-order interactions.

**Generalizability of interactions.** Given a target LLM $v$ and an input prompt $\boldsymbol{x}$, we follow (Chen et al., 2024) to quantify the generalizability of the interactions encoded by $v$. An interaction is considered generalizable if it is consistently extracted across different DNNs. To this end, we introduce a baseline LLM $v'$ whose architecture differs from that of the target model. Specifically, if an interaction $S$ encoded by the target LLM $v$ is also captured by the baseline LLM $v'$, then this interaction $S$ is regarded as generalizable. Thus, the generalizability of an interaction $S$ can be quantified using the following binary metric.

$$\forall \text{type} \in \{\text{and, or}\}, \forall S \in \Omega^{\text{type}},\ \mathcal{G}_S^{\text{type}} \overset{\text{def}}{=} \mathbb{1}(S \in \Omega_{v'}^{\text{type}}) \cdot \mathbb{1}(\text{sign}(I_S^{\text{type}}) = \text{sign}(I_{S,v'}^{\text{type}})), \tag{7}$$

where $\mathbb{1}(\cdot) \in \{0, 1\}$ is a trigger function that returns 1 if the given condition is satisfied. $\Omega_{v'}^{\text{and}}$ and $\Omega_{v'}^{\text{or}}$ denote the sets of AND and OR interactions encoded by the baseline LLM $v'$. $I_{S,v'}^{\text{and}}$ and $I_{S,v'}^{\text{or}}$ denote the scalar interaction effects encoded by the baseline LLM $v'$. $\text{sign}(I_S^{\text{type}}) = \text{sign}(I_{S,v'}^{\text{type}})$ indicates that interaction $S$ has a consistent effect on the prediction of the target token, meaning that it either increases or decreases the prediction score of the target token in both LLMs.

**Distribution of interactions.** Instead of analyzing the complexity of each interaction individually, we characterize the distribution of interactions over different orders. For example, the distribution of newly emerged positive interactions over different orders and that of newly emerged negative interactions are represented by two vectors, $\boldsymbol{e}^+ = [\mathrm{e}^{(1),+}, \mathrm{e}^{(2),+}, \cdots, \mathrm{e}^{(n),+}]^T$ and $\boldsymbol{e}^- = [\mathrm{e}^{(1),-}, \mathrm{e}^{(2),-}, \cdots, \mathrm{e}^{(n),-}]^T$, respectively. $\mathrm{e}^{(k),+}$ and $\mathrm{e}^{(k),-}$ aggregate all emerged positive interactions of the $k$-th order and all emerged negative interactions of the $k$-th order, respectively.

$$\begin{aligned}
\mathrm{e}^{(k),+} &= f_{\text{strength}}^{(k),+}(\boldsymbol{E}_t^{\text{and}}, \boldsymbol{E}_t^{\text{or}}) = \sum\nolimits_{\text{type}\in\{\text{and,or}\}} \sum\nolimits_{S\in\boldsymbol{E}_t^{\text{type}}:|S|=k} \max(I_S^{\text{type}}, 0), \\
\mathrm{e}^{(k),-} &= f_{\text{strength}}^{(k),-}(\boldsymbol{E}_t^{\text{and}}, \boldsymbol{E}_t^{\text{or}}) = \sum\nolimits_{\text{type}\in\{\text{and,or}\}} \sum\nolimits_{S\in\boldsymbol{E}_t^{\text{type}}:|S|=k} \min(I_S^{\text{type}}, 0).
\end{aligned} \tag{8}$$

Similarly, the distributions of removed interactions are denoted by $\boldsymbol{r}^+ = [\mathrm{r}^{(1),+}, \mathrm{r}^{(2),+}, \cdots, \mathrm{r}^{(n),+}]^T$ and $\boldsymbol{r}^- = [\mathrm{r}^{(1),-}, \mathrm{r}^{(2),-}, \cdots, \mathrm{r}^{(n),-}]^T$, respectively[5]. The distributions of preserved interactions are represented by $\boldsymbol{p}^+ = [\mathrm{p}^{(1),+}, \mathrm{p}^{(2),+}, \cdots, \mathrm{p}^{(n),+}]^T$ and $\boldsymbol{p}^- = [\mathrm{p}^{(1),-}, \mathrm{p}^{(2),-}, \cdots, \mathrm{p}^{(n),-}]^T$, respectively[5].

### 3.2.2 The Role of SFT: Denoising Rather than Reliable Representation Learning

In this subsection, we conduct experiments to track the evolution of newly emerged, removed, and preserved interactions across the SFT process. We find that (1) LLMs primarily remove noise-like interactions during SFT, while rarely learning reliable interactions; (2) this denoising stage is extremely brief, after which continued fine-tuning leads to learning numerous overfitted patterns.

*Experimental setting.* We perform SFT on multiple LLMs across different datasets. For the GoEmotions dataset (Demszky et al., 2020), we fine-tune Qwen2.5-3B-Instruct (Team, 2024), Qwen2.5-7B-Instruct (Team, 2024), Llama-2-7B-Chat (Touvron et al., 2023), and Llama-3-8B-Instruct (Grattafiori et al., 2024). For the Unilaw-R1-Data dataset (Cai et al., 2025), we fine-tune Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. For the Databricks-Dolly-15k dataset (Conover et al., 2023), we fine-tune Gemma-3-4B-it (Team et al., 2025). All fine-tuning experiments are conducted using LoRA, a parameter-efficient fine-tuning method (Hu et al., 2021). Please see Appendix E.1 for detailed settings and dataset descriptions. Moreover, we follow (Chen et al., 2024) to sample a set of words from the input prompt, where the embedding of each sampled word is treated as an input variable. Please see Appendix E.4 for details.

[5]Elements in $\boldsymbol{r}^+$ and $\boldsymbol{r}^-$ are defined as $\mathrm{r}^{(k),+} = f_{\text{strength}}^{(k),+}(\boldsymbol{R}_t^{\text{and}}, \boldsymbol{R}_t^{\text{or}})$ and $\mathrm{r}^{(k),-} = f_{\text{strength}}^{(k),-}(\boldsymbol{R}_t^{\text{and}}, \boldsymbol{R}_t^{\text{or}})$, respectively. Similarly, elements in $\boldsymbol{p}^+$ and $\boldsymbol{p}^-$ are defined as $\mathrm{p}^{(k),+} = f_{\text{strength}}^{(k),+}(\boldsymbol{P}_t^{\text{and}}, \boldsymbol{P}_t^{\text{or}})$ and $\mathrm{p}^{(k),-} = f_{\text{strength}}^{(k),-}(\boldsymbol{P}_t^{\text{and}}, \boldsymbol{P}_t^{\text{or}})$.

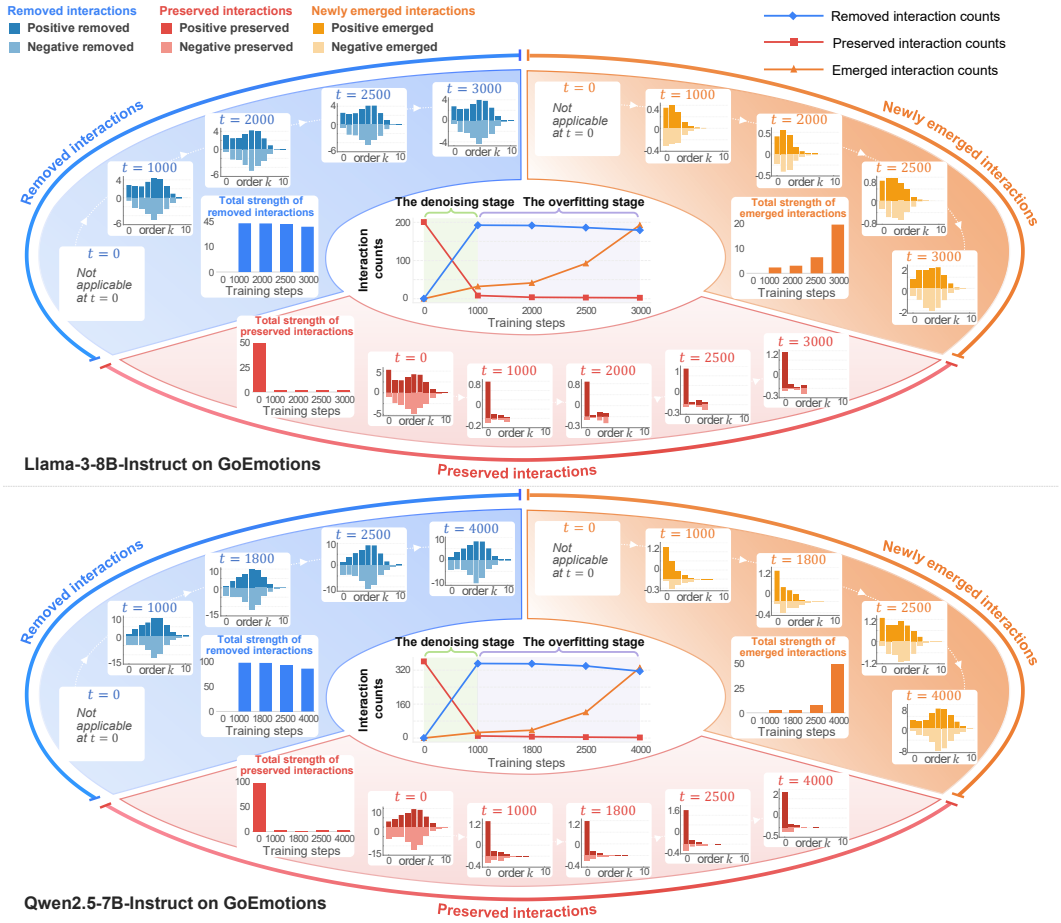


Figure 2: Evolution of the distribution of newly emerged interactions (*i.e.,* $\boldsymbol{e}^+$ and $\boldsymbol{e}^-$), removed interactions (*i.e.,* $\boldsymbol{r}^+$ and $\boldsymbol{r}^-$), and preserved interactions (*i.e.,* $\boldsymbol{p}^+$ and $\boldsymbol{p}^-$) throughout the SFT process. In the very short denoising stage of SFT, the LLM removes a large number of mutually canceling interactions, while only a small number of low-order interactions are preserved. In the later overfitting stage, many high-order mutually canceling interactions gradually emerge again. Additional experimental results on more LLMs are shown in Appendix D.1.

There are two commonly used metrics for evaluating the representation quality of interactions.

**Metric 1 (Ratio of generalizable interactions).** *Given a set of AND-OR interactions* $\Omega^{and}$ *and* $\Omega^{or}$*, we quantify the proportion of generalizable interactions among all interactions,* $\gamma(\Omega^{and}, \Omega^{or}) \in [0, 100\%]$*, as follows.*

$$\gamma(\Omega^{and}, \Omega^{or}) \overset{def}{=} \frac{\sum_{type\in\{and,or\}} \sum_{S\in\Omega^{type}} |I_S^{type} \cdot \mathcal{G}_S^{type}|}{\sum_{type\in\{and,or\}} \sum_{S\in\Omega^{type}} |I_S^{type}|} \times 100\%, \tag{9}$$

*where* $\mathcal{G}_S^{type} \in \{0, 1\}$*, defined in Equation (7), quantifies the generalizability of the interaction* $S$*.*

**Metric 2 (Ratio of uncancelled interaction effects).** *Given a set of AND-OR interactions* $\Omega^{and}$ *and* $\Omega^{or}$*, we quantify the ratio of interaction effects that remain uncancelled after the cancellation between positive and negative effects by* $\rho(\Omega^{and}, \Omega^{or}) \in [0, 100\%]$*, as follows.*

$$\rho(\Omega^{and}, \Omega^{or}) \overset{def}{=} \frac{\left|\sum_{type\in\{and,or\}} \sum_{S\in\Omega^{type}} I_S^{type}\right|}{\sum_{type\in\{and,or\}} \sum_{S\in\Omega^{type}} |I_S^{type}|} \times 100\%. \tag{10}$$

If $\rho(\Omega^{\text{and}}, \Omega^{\text{or}})$ is close to zero, it means that there is strong cancellation between positive and negative effects, suggesting that these interactions make little contribution to the prediction of target tokens and are more likely to reflect noise patterns.

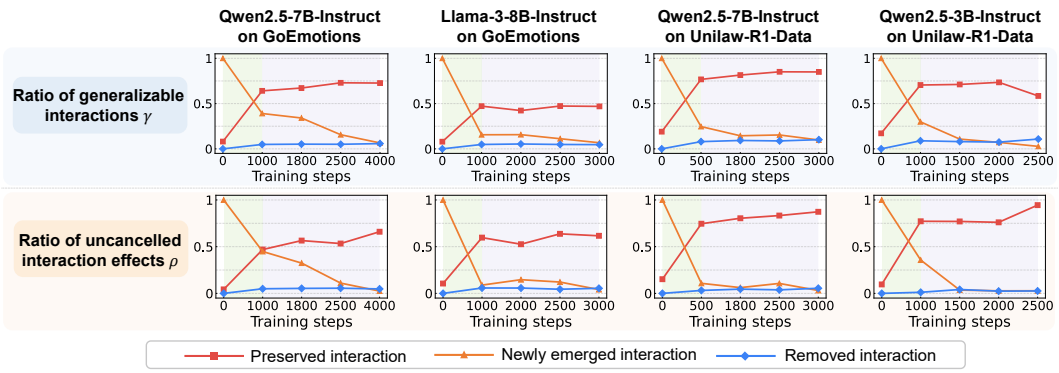


Figure 3: Evolution of the representation quality of newly emerged, removed, and preserved interactions during SFT, measured by generalizability $\gamma$ and uncancelled-effect ratio $\rho$. Newly emerged interactions show high quality only in the short denoising stage (the green region in the figure), but become less generalizable and more mutually canceling later (the purple region). Removed interactions remain low on both metrics throughout SFT, while preserved interactions improve after noise interactions are removed. Additional experimental results on more LLMs are shown in Appendix D.2.

**(1) Analyzing newly emerged interactions.** We track and analyze the representation quality of newly emerged interactions at different time points during SFT (*e.g.,* the ratio of generalizable newly emerged interactions $\gamma(\boldsymbol{E}_t^{\text{and}}, \boldsymbol{E}_t^{\text{or}})$ at time point $t$). Figure 2 shows that the learning of newly emerged interactions during the SFT process can be divided into two stages.

**Finding 1:** *LLMs learn only a few newly emerged interactions in the first stage, but begin to learn a large number of such interactions in the second stage.* Specifically, Figure 2 compares the interaction counts, $\text{count}(\boldsymbol{E}_t^{\text{and}}, \boldsymbol{E}_t^{\text{or}}) = |\boldsymbol{E}_t^{\text{and}}| + |\boldsymbol{E}_t^{\text{or}}|$, and the total interaction strength, $\sum_{k=1}^{n}(|\text{e}^{(k),+}| + |\text{e}^{(k),-}|)$, of interactions emerging in the first stage with those emerging in the second stage. The results show that only a small proportion of interactions emerge in the first denoising stage (green area in the figure), whereas the vast majority emerge in the second overfitting stage (purple area in the figure).

**Finding 2:** *The small number of interactions that emerge in the first stage tend to be more generalizable, whereas the large number of interactions that emerge in the second stage behave more like noise patterns.* This claim is supported by three observations. (i) Figure 2 shows that interactions emerging in the second stage (*i.e.,* the overfitting stage, corresponding to the purple region) are mostly high-order interactions, which are generally less reliable than those emerging in the first stage (*i.e.,* the denoising stage, corresponding to the green region). (ii) Figure 3 shows that interactions emerging in the first stage are more generalizable than those emerging in the second stage, indicating that most interactions emerging in the second stage are model-specific rather than shared across different DNNs. (iii) Figure 3 further shows that interactions emerging in the first stage have a much higher ratio of uncancelled interaction effects than those emerging in the second stage.

These results suggest that LLMs tend to learn only a limited number of relatively reliable interactions in the first stage, while the huge number of newly emerged interactions in later stages are less likely to generalize and act like noise.

**(2) Analyzing removed interactions.** Figure 2 shows that the removal of interactions during the SFT process can also be divided into two stages.

**Finding 3:** *The interaction removal in LLMs occurs primarily within an extremely short first stage.* The LLM removes the vast majority of interactions during the first stage, while only a negligible amount is removed in the second stage. Figure 2 compares the interaction counts, $\text{count}(\boldsymbol{R}_t^{\text{and}}, \boldsymbol{R}_t^{\text{or}}) = |\boldsymbol{R}_t^{\text{and}}| + |\boldsymbol{R}_t^{\text{or}}|$, and the total interaction strength, $\sum_{k=1}^{n}(|\text{r}^{(k),+}| + |\text{r}^{(k),-}|)$, of interactions removed in the first stage with those removed in the second stage. The results show that most interaction removal occurs in the first stage, with only very few interactions being removed in the second stage.

**Finding 4:** *The interactions removed during SFT predominantly correspond to noise patterns with poor generalizability.* This claim is supported by three observations. (i) Figure 2 shows that removed interactions are mostly high-order interactions, suggesting that they encode complex and less reliable patterns. (ii) Figure 3 shows that almost none of the removed interactions are generalizable across different LLMs ($\gamma(\boldsymbol{R}_t^{\text{and}}, \boldsymbol{R}_t^{\text{or}}) \approx 0$), indicating that they are largely model-specific. (iii) Figure 3

further shows that nearly all removed interaction effects exhibit mutual cancellation ($\rho(\boldsymbol{R}_t^{\text{and}}, \boldsymbol{R}_t^{\text{or}}) \approx 0$), suggesting that they contribute only marginally to target token prediction.

These results highlight one of the beneficial effects of SFT, *i.e.,* it effectively removes unreliable/noise representations. However, this denoising process is short-lived, and continued fine-tuning beyond this stage does not further strengthen this effect.

**(3) Analyzing preserved interactions.** Figure 2 shows that the set of preserved interactions gradually stabilizes within the first $1000$ training steps. Afterward, the composition of these interactions remains largely unchanged, whereas their strengths continue to be reinforced during training.

**Finding 5:** *LLMs retain only a small set of interactions during SFT, and these preserved interactions exhibit substantially higher representation quality than the other two types of interactions.* This claim is supported by three observations. (i) Figure 2 shows that preserved interactions are predominantly low-order interactions, suggesting that they represent simple and reliable inference patterns. (ii) Figure 3 shows that, in most cases, more than half of the preserved interactions are generalizable across different LLMs, *i.e.,* $\gamma(\boldsymbol{P}_t^{\text{and}}, \boldsymbol{P}_t^{\text{or}}) > 50\%$, indicating that they capture shared representations across LLMs. (iii) Figure 3 further shows that preserved interactions exhibit weaker positive-negative cancellation, *i.e.,* larger $\rho$ values, than the other two types of interactions, suggesting that they contribute more consistently to target token prediction. These results suggest that one of the beneficial effects of SFT is to identify and reinforce high-quality interactions that already exist in the LLM.

**The primary utility of SFT lies in rapid denoising, followed by prolonged overfitting.** In summary, all these results suggest that during SFT, LLMs primarily remove noise-like interactions without learning many reliable interactions. Moreover, this denoising stage is extremely short. Once this stage ends, continued fine-tuning mainly introduces numerous noise-like interactions. *We further find that downstream task performance is temporally aligned with these two stages: it improves substantially during the short denoising stage, but no longer improves and often declines during the subsequent stage. Detailed results are provided in Appendix D.3.*

### 3.3 Are Preserved Interactions the Backbone of LLM Inference?

Since our previous findings suggest that the primary utility of SFT lies in denoising rather than learning new interactions, in this subsection, we further examine the following hypothesis: **the small set of interactions preserved in SFT constitutes the key backbone supporting target token prediction in LLMs**. In other words, we ask whether most interaction effects used for token prediction originate from this small set of preserved interactions, while the large number of removed interactions and interactions newly emerged in later stages contribute only marginally to the model's predictions. We validate this hypothesis from the following three perspectives.

**(1) Verifying that removed and newly emerged interactions are predominantly noise, which is characterized by positive-negative cancellation of interaction effects.** We measure the ratio of uncancelled interaction effects $\rho$ (see Equation (10)) of different types of interactions, including preserved, newly emerged, and removed interactions.

Figure 3 shows that preserved interactions exhibit the highest ratio of uncancelled effects, whereas newly emerged and removed interactions are largely characterized by positive-negative cancellation. This suggests that the latter predominantly capture noise patterns, while preserved interactions encode meaningful signals that contribute to prediction.

**(2) Verifying that preserved interactions exhibit a larger average contribution of individual interactions to target token prediction.** We measure the average contribution of individual removed interactions[6] as $\sum_{k=1}^{n}(\mathrm{r}^{(k),+} + \mathrm{r}^{(k),-})/(|\boldsymbol{R}_t^{\text{and}}| + |\boldsymbol{R}_t^{\text{or}}|)$. The corresponding quantities for newly emerged[6] and preserved interactions are computed as $\sum_{k=1}^{n}(\mathrm{e}^{(k),+} + \mathrm{e}^{(k),-})/(|\boldsymbol{E}_t^{\text{and}}| + |\boldsymbol{E}_t^{\text{or}}|)$ and $\sum_{k=1}^{n}(\mathrm{p}^{(k),+} + \mathrm{p}^{(k),-})/(|\boldsymbol{P}_t^{\text{and}}| + |\boldsymbol{P}_t^{\text{or}}|)$, respectively.

Figure 4 shows that preserved interactions and early-emerged interactions in the denoising stage contribute substantially more to target token prediction than removed interactions and later-emerged interactions in the overfitting stage. In contrast, the weak contributions of removed and later-emerged interactions suggest that they are more likely to represent noise patterns.

[6]Preserved and removed interactions are identified from the LLM after SFT. For newly emerged interactions, we separately evaluate interactions emerging during the denoising stage and those in the subsequent stage.

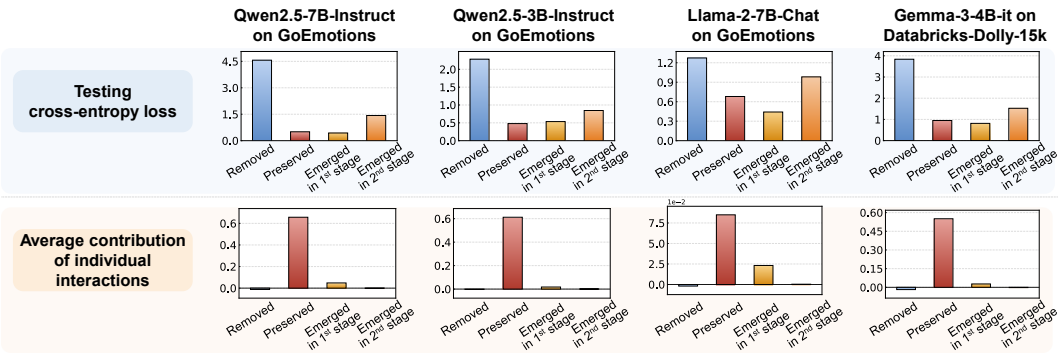


Figure 4: Prediction utility and individual contributions of different types of interactions. We compare removed interactions, preserved interactions, early-emerged interactions in the denoising stage, and later-emerged interactions in the overfitting stage. Preserved and early-emerged interactions yield lower test loss and larger per-interaction contributions, whereas removed and later-emerged interactions contribute only marginally. These results suggest that preserved interactions, together with a few early-emerged interactions, form the main backbone of LLM inference.

**(3) Verifying that preserved interactions are sufficient for basic inference.** We evaluate the contribution of different types of interactions to target token prediction. In the setting of single-token generation, the prediction probability of the target token $y^*$ can be recovered from the output score as $p(y^*|\boldsymbol{x}) = g^{-1}(v(\boldsymbol{x}))$[7], according to Equation (1). Because the output score of an LLM can be decomposed into the sum of all interaction effects (see Equation (3)), the following probability naturally measures the utility of using only preserved interactions for token prediction.

$$p_{\text{preserved}}(y^*|\boldsymbol{x}) = g^{-1}(\sum_{k=1}^{n} \mathrm{p}^{(k),+} + \mathrm{p}^{(k),-}). \tag{11}$$

Similarly, $p_{\text{removed}}(y^*|\boldsymbol{x}) = g^{-1}(\sum_{k=1}^{n} \mathrm{r}^{(k),+} + \mathrm{r}^{(k),-})$ and $p_{\text{emerged}}(y^*|\boldsymbol{x}) = g^{-1}(\sum_{k=1}^{n} \mathrm{e}^{(k),+} + \mathrm{e}^{(k),-})$ measure the utilities of using only removed interactions and newly emerged interactions for token prediction, respectively.

Figure 4 compares the cross-entropy loss on test samples when target token prediction is performed using only preserved interactions, removed interactions, early-emerged interactions[6] in the denoising stage, or later-emerged interactions in the overfitting stage. Specifically, the losses are computed from the prediction probabilities induced by each type of interaction[6]. The results show that using only preserved interactions or early-emerged interactions yields lower loss than using only removed interactions or later-emerged interactions. This indicates that preserved interactions and a few early-emerged interactions form the backbone supporting target token prediction.

Taken together, the above three sets of experiments provide consistent evidence that preserved interactions form the core backbone underlying target token prediction in LLMs.

## 4 Conclusion and Discussion

In this paper, we use interactions to address the scientific question of the inconsistent effectiveness of SFT on LLMs. We find that: (1) SFT primarily removes noise-like interactions from LLMs, while rarely inducing reliable new interactions; and (2) this denoising stage is extremely brief, after which continued fine-tuning leads to the emergence of numerous overfitted patterns. We validate these findings across multiple LLMs and datasets. These results help explain why SFT can sometimes exhibit inconsistent or even detrimental effects when applied to LLMs.

Our findings suggest that interactions may provide diagnostic signals for monitoring SFT and provide a more principled criterion for early stopping in end-to-end SFT. In practice, our study challenges the conventional belief that fine-tuning LLMs on massive datasets is necessarily beneficial. The brief denoising stage suggests limited benefits from continued fine-tuning after denoising. Moreover, our method provides a way to identify the early stopping point for a given LLM.

[7]The model output score on an input $\boldsymbol{x}$ can be written as $v(\boldsymbol{x}) = g(p(y^*|\boldsymbol{x})) = \log(p(y^*|\boldsymbol{x})/(1 - p(y^*|\boldsymbol{x})))$.

## AI USE STATEMENT

Generative AI tools were used solely to improve the clarity, grammar, and readability of the manuscript. They were not used to develop research ideas, theoretical analyses, methods, experiments, data analysis, result interpretation, or scientific conclusions. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content of this work.

## REPRODUCIBILITY STATEMENT

Detailed descriptions of the datasets, models, fine-tuning configurations, and interaction computation are provided in Appendix E. The complete procedure and pseudocode for extracting AND-OR interactions are presented in Appendix F. We will publicly release the source code and configuration files upon acceptance.

## REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Landon Butler, Abhineet Agarwal, Justin Singh Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. Proxyspex: Inference-efficient interpretability via sparse feature interactions in llms. *arXiv preprint arXiv:2505.17495*, 2025.

Hua Cai, Shuang Zhao, Liang Zhang, Xuli Shen, Qing Xu, Weilin Shen, Zihao Wen, and Tianke Ban. Unilaw-r1: A large language model for legal reasoning with reinforcement learning and iterative inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18128–18142, 2025.

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. URL https://openreview.net/forum?id=OCqyFVFNeF.

Lu Chen, Yuxuan Huang, Yixing Li, Dongrui Liu, Qihan Ren, Kun Kuang, Zilong Zheng, Quanshi Zhang, et al. Can llms reason soundly in law? auditing inference patterns for legal judgment. In *The Fourteenth International Conference on Learning Representations*, 2026.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4040–4054, 2020.

Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. Discovering and explaining the representation bottleneck of dnns. *arXiv preprint arXiv:2111.06236*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Justin S Kang, Yigit E Erginbas, Landon Butler, Ramtin Pedarsani, and Kannan Ramchandran. Learning to understand: Identifying interactions via the möbius transform. *Advances in Neural Information Processing Systems*, 37:46160–46202, 2024.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Kannan Ramchandran, and Bin Yu. Spex: Scaling feature interaction explanations for llms. *arXiv preprint arXiv:2502.13870*, 2025.

Mingjie Li and Quanshi Zhang. Defining and quantifying and-or interactions for faithful and concise explanation of dnns. *arXiv preprint arXiv:2304.13312*, 2023a.

Mingjie Li and Quanshi Zhang. Does a neural network really encode symbolic concepts? In *International conference on machine learning*, pp. 20452–20469, 2023b.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in dnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20280–20289, 2023a.

Qihan Ren, Huiqi Deng, Yunuo Chen, Siyu Lou, and Quanshi Zhang. Bayesian neural networks avoid encoding complex and perturbation-sensitive concepts. In *International Conference on Machine Learning*, pp. 28889–28913. PMLR, 2023b.

Qihan Ren, Jiayang Gao, Wen Shen, and Quanshi Zhang. Where we have arrived in proving the emergence of sparse interaction primitives in dnns. In *The Twelfth International Conference on Learning Representations*, 2024a.

Qihan Ren, Junpeng Zhang, Yang Xu, Yue Xin, Dongrui Liu, and Quanshi Zhang. Towards the dynamics of a dnn learning symbolic interactions. *Advances in Neural Information Processing Systems*, 37:50653–50688, 2024b.

Lloyd S Shapley et al. A value for n-person games. 1953.

Zhengyan Shi, Adam X Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, and Aldo Lipani. Instruction tuning with loss over instructions. *Advances in Neural Information Processing Systems*, 37:69176–69205, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

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*, 2023.

Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. *arXiv preprint arXiv:2010.04055*, 2020.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. Two-stage llm fine-tuning with less specialization and more generalization. *arXiv preprint arXiv:2211.00635*, 2022.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pp. 400–410, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Explaining generalization power of a dnn using interactive concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17105–17113, 2024.

Huilin Zhou, Qihan Ren, Junpeng Zhang, and Quanshi Zhang. Towards the first principles of explaining dnns: interactions explain the learning dynamics. *Frontiers of Information Technology & Electronic Engineering*, 26(7):1017–1026, 2025.

# APPENDIX

This appendix provides detailed information that supports the main paper. For clarity, the appendix is organized as follows.

- Appendix A provides validation of the high fidelity of interaction-based explanations.
- Appendix B provides validation of the sparsity of interactions.
- Appendix C discusses the computational cost and practical scalability of interactions.
- Appendix D reports additional experimental results.
- Appendix E presents experimental details.
- Appendix F describes the procedure for extracting the sparsest AND-OR interactions.
- Appendix G discusses the practical implications of interaction analysis for monitoring SFT and early stopping.

## A VALIDATION OF HIGH-FIDELITY INTERACTION-BASED EXPLANATION

In this section, we validate the high fidelity of the interaction-based explanation from both theoretical and empirical perspectives. Specifically, Chen et al. (2024) have theoretically proven that the logical model $\phi$, constructed from the extracted AND-OR interactions, can accurately predict the LLM outputs for all $2^n$ masked states of $\boldsymbol{x}$. Therefore, for completeness, we first review this existing theoretical guarantee to establish the fidelity of $\phi$, and then further corroborate it through our empirical validation.

### A.1 THEORETICAL VALIDATION

*Proof.* **(1) Universal matching theorem of AND interactions.**

We will prove that output component $u_S^{\text{and}}$ on all $2^n$ masked samples $\{\boldsymbol{x}_S : S \subseteq N\}$ could be universally explained by all interactions in $S \subseteq N$, *i.e.*, $\forall \emptyset \neq S \subseteq N, u_S^{\text{and}} = \sum_{\emptyset \neq T \subseteq S} I_T^{\text{and}} + v(\boldsymbol{x}_\emptyset)$. In particular, we define $v_\emptyset^{\text{and}} = v(\boldsymbol{x}_\emptyset)$ (*i.e.*, we attribute output on an empty sample to AND interactions).

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

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

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

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

$$
\begin{aligned}
\sum\nolimits_{\emptyset \neq T \subseteq S} I_T^{\text{and}} &= \sum\nolimits_{\emptyset \neq T \subseteq S} \sum\nolimits_{L \subseteq T} (-1)^{|T|-|L|} u_L^{\text{and}} \\
&= \sum\nolimits_{L \subseteq S} \sum\nolimits_{T: L \subseteq T \subseteq S} (-1)^{|T|-|L|} u_L^{\text{and}} - v_\emptyset^{\text{and}} \\
&= \underbrace{u_S^{\text{and}}}_{L=S} + \sum\nolimits_{L \subseteq S, L \neq S} u_L^{\text{and}} \cdot \underbrace{\sum\nolimits_{m=0}^{|S|-|L|} C_{|S|-|L|}^m (-1)^m}_{=0} - v_\emptyset^{\text{and}} \\
&= u_S^{\text{and}} - v_\emptyset^{\text{and}} = u_S^{\text{and}} - v(\boldsymbol{x}_\emptyset)
\end{aligned}
\tag{12}
$$

Thus, we have $\forall \emptyset \neq S \subseteq N, u_S^{\text{and}} = \sum_{\emptyset \neq T \subseteq S} I_T^{\text{and}} + v(\boldsymbol{x}_\emptyset)$.

**(2) Universal matching theorem of OR interactions.**

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

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

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

(1) When $L = N \setminus S$, the linear combination of all subsets $T$ containing $L$ with respect to the model output $v^{\text{or}}_{N\setminus L}$ is $\sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v^{\text{or}}_{N\setminus L} = \sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}u^{\text{or}}_{S}$. For all sets $T : T \supseteq L, T \cap S \neq \emptyset$ (then $T \neq N \setminus S, T \neq L$), let us consider the linear combinations of all sets $T$ with number $|T|$ for the model output $u^{\text{or}}_S$, respectively. Let $|T'| := |T| - |L|$, $(1 \leq |T'| \leq |S|)$, then there are a total of $C^{|T'|}_{|S|}$ combinations of all sets $T'$ of order $|T'|$. Thus, given $L$, accumulating the model outputs $u^{\text{or}}_S$ corresponding to all $T \supseteq L$, then $\sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v^{\text{or}}_{N\setminus L} = u^{\text{or}}_S \cdot \underbrace{\sum_{|T'|=1}^{|S|} C^{|T'|}_{|S|}(-1)^{|T'|}}_{=-1} = -u^{\text{or}}_S$.

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

(3) When $L \cap S \neq \emptyset, L \neq N$, the linear combination of all subsets $T$ containing $L$ with respect to the model output $v^{\text{or}}_{N\setminus L}$ is $\sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v^{\text{or}}_{N\setminus L}$. For all sets $T : T \supseteq L, T \cap S \neq \emptyset$, let us consider the linear combinations of all sets $T$ with number $|T|$ for the model output $u^{\text{or}}_S$, respectively. Let us split $|T| - |L|$ into $|T'|$ and $|T''|$, *i.e.*, $|T| - |L| = |T'| + |T''|$, where $T' = \{i|i \in T, i \notin L, i \in N \setminus S\}$, $T'' = \{i|i \in T, i \notin L, i \in S\}$ (then $0 \leq |T''| \leq |S| - |S \cap L|$) and $|T'| + |T''| + |L| = |T|$. In this way, there are a total of $C^{|T''|}_{|S|-|S\cap L|}$ combinations of all sets $T''$ of order $|T''|$. Thus, given $L$, accumulating the model outputs $v^{\text{or}}_{N\setminus L}$ corresponding to all $T \supseteq L$, then $\sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v^{\text{or}}_{N\setminus L} = v^{\text{or}}_{N\setminus L} \cdot \sum_{T'\subseteq N\setminus S\setminus L} \underbrace{\sum_{|T''|=0}^{|S|-|S\cap L|} C^{|T''|}_{|S|-|S\cap L|}(-1)^{|T'|+|T''|}}_{=0} = 0$.

(4) When $L \cap S = \emptyset, L \neq N \setminus S$, the linear combination of all subsets $T$ containing $L$ with respect to the model output $v^{\text{or}}_{N\setminus L}$ is $\sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v^{\text{or}}_{N\setminus L}$. Similarly, let us split $|T| - |L|$ into $|T'|$ and $|T''|$, *i.e.*, $|T| - |L| = |T'| + |T''|$, where $T' = \{i|i \in T, i \notin L, i \in N \setminus S\}$, $T'' = \{i|i \in T, i \in S\}$ (then $0 \leq |T''| \leq |S|$) and $|T'| + |T''| + |L| = |T|$. In this way, there are a total of $C^{|T''|}_{|S|}$ combinations of all sets $T''$ of order $|T''|$. Thus, given $L$, accumulating the model outputs $v^{\text{or}}_{N\setminus L}$ corresponding to all $T \supseteq L$, then $\sum_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v^{\text{or}}_{N\setminus L} = v^{\text{or}}_{N\setminus L} \cdot \sum_{T'\subseteq N\setminus S\setminus L} \underbrace{\sum_{|T''|=0}^{|S|} C^{|T''|}_{|S|}(-1)^{|T'|+|T''|}}_{=0} = 0$.

Please see the complete derivation of the following formula.

$$
\begin{aligned}
\sum\nolimits_{T:T\cap S\neq\emptyset} I_T^{\text{or}} &= \sum\nolimits_{T:T\cap S\neq\emptyset}\left[-\sum\nolimits_{L\subseteq T}(-1)^{|T|-|L|}v_{N\setminus L}^{\text{or}}\right] \\
&= -\sum\nolimits_{L\subseteq N}\sum\nolimits_{T:T\cap S\neq\emptyset,T\supseteq L}(-1)^{|T|-|L|}v_{N\setminus L}^{\text{or}} \\
&= -\left[\sum_{|T'|=1}^{|S|} C_{|S|}^{|T'|}(-1)^{|T'|}\right]\cdot \underbrace{u_S^{\text{or}}}_{L=N\setminus S} - \underbrace{v_\emptyset^{\text{or}}}_{L=N} \\
&\quad - \sum_{L\cap S\neq\emptyset,L\neq N}\left[\sum_{T'\subseteq N\setminus S\setminus L}\left(\sum_{|T''|=0}^{|S|-|S\cap L|} C_{|S|-|S\cap L|}^{|T''|}(-1)^{|T'|+|T''|}\right)\right]\cdot v_{N\setminus L}^{\text{or}} \\
&\quad - \sum_{L\cap S=\emptyset,L\neq N\setminus S}\left[\sum_{T'\subseteq N\setminus S\setminus L}\left(\sum_{|T''|=0}^{|S|} C_{|S|}^{|T''|}(-1)^{|T'|+|T''|}\right)\right]\cdot v_{N\setminus L}^{\text{or}} \\
&= -(-1)\cdot u_S^{\text{or}} - v_\emptyset^{\text{or}} - \sum_{L\cap S\neq\emptyset,L\neq N}\left[\sum_{T'\subseteq N\setminus S\setminus L} 0\right]\cdot v_{N\setminus L}^{\text{or}} \\
&\quad - \sum_{L\cap S=\emptyset,L\neq N\setminus S}\left[\sum_{T'\subseteq N\setminus S\setminus L} 0\right]\cdot v_{N\setminus L}^{\text{or}} \\
&= u_S^{\text{or}} - v_\emptyset^{\text{or}} \\
&= u_S^{\text{or}}
\end{aligned}
\tag{13}
$$

**(3) Universal matching theorem of AND-OR interactions.** With the universal matching theorem of AND interactions and the universal matching theorem of OR interactions, we can easily get $v(\boldsymbol{x}_S) = u_S^{\text{and}} + u_S^{\text{or}} = v(\boldsymbol{x}_\emptyset) + \sum_{\emptyset\neq T\subseteq S} I_T^{\text{and}} + \sum_{T:T\cap S\neq\emptyset} I_T^{\text{or}}$, thus, we obtain the universal matching theorem of AND-OR interactions.

□

### A.2 Empirical Verification of Universal Matching on LLMs

Given an input prompt $\boldsymbol{x}$, the above subsection theoretically shows that the LLM output function $v(\cdot)$ can be faithfully matched by the corresponding logical model function $\phi(\cdot)$ over all $2^n$ different masked states in $\Psi$. In this subsection, we further provide empirical evidence to verify this universal matching property on large language models.

Specifically, for each input sample $\boldsymbol{x}$, we compute the outputs of both the LLM and the corresponding logical model on all masked states $\boldsymbol{x}_S \in \Psi$. We then sort these masked states according to the LLM output $v(\boldsymbol{x}_S)$ in descending order, and compare the logical model output $\phi(\boldsymbol{x}_S)$ at the corresponding positions. As shown in Figure 5, the logical model closely matches the output curve of the LLM across different masked states, indicating that the logical model can accurately approximate the behavior of the LLM function.

In addition, we define the matching error as

$$
\epsilon(\boldsymbol{x}_S) = \frac{\phi(\boldsymbol{x}_S) - v(\boldsymbol{x}_S)}{|v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)|}, \tag{14}
$$

where the denominator $|v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)|$ normalizes the error by the output scale of the LLM on the given sample. Figure 5 shows the histogram of matching errors over different masked states and different input samples. The distribution is highly concentrated around zero, demonstrating that the matching error remains very small across samples and masking states. These empirical results further verify the high fidelity of the logical model in approximating LLM outputs.

We conduct experiments on multiple LLMs and datasets, including Llama2-7B-Chat and Qwen2.5-7B-Instruct on the GoEmotions dataset, Qwen2.5-3B-Instruct on the Unilaw-R1-Data dataset, and

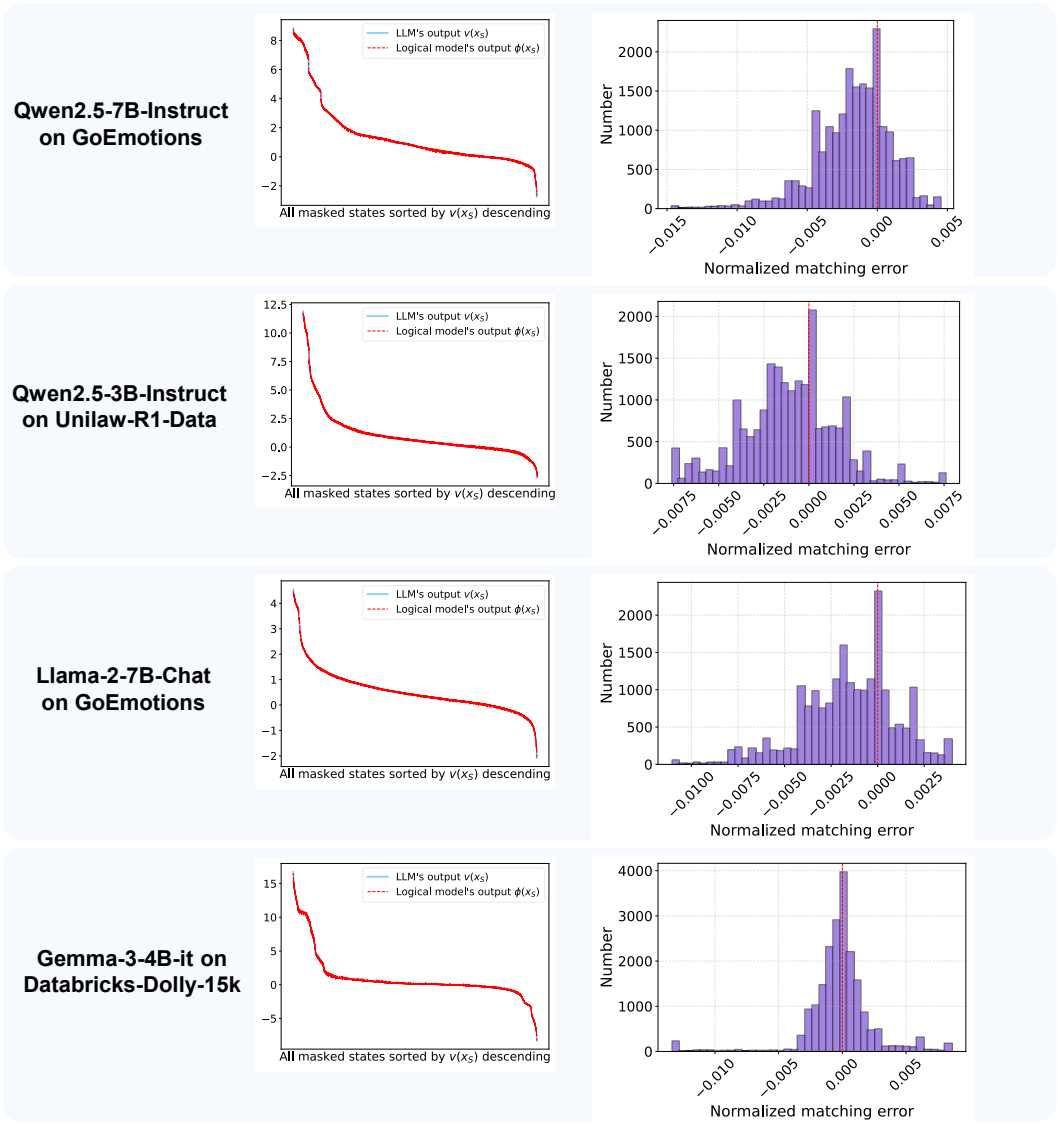


Figure 5: Empirical verification of universal matching on LLMs. Each row corresponds to a different model-dataset setting. In the left column, we aggregate the masked states from all selected samples and sort them in descending order according to the LLM output $v(\boldsymbol{x}_S)$. The logical model output $\phi(\boldsymbol{x}_S)$ is plotted at the corresponding sorted positions, showing that the logical model closely matches the LLM output across different masked states. In the right column, we report the histogram of the normalized matching error $\epsilon(\boldsymbol{x}_S) = (\phi(\boldsymbol{x}_S) - v(\boldsymbol{x}_S))/|v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)|$ over different masked states and input samples. The errors are concentrated around zero across different model-dataset settings, further demonstrating the high fidelity of the logical model in approximating LLM outputs.

Gemma-3-4B-it on the Databricks-Dolly-15k dataset. For each model-dataset setting, we compute interaction-based logical models and compare their outputs with the corresponding LLM outputs over all masked states of the selected samples. The results consistently show that the logical models can faithfully reproduce the LLM output functions, providing strong empirical evidence for the universal matching ability of our interaction-based explanations.

# B VALIDATION OF THE SPARSITY OF INTERACTIONS

Ren et al. (2024a) have proved three sufficient conditions for the sparsity of AND interactions. Accordingly, in this section, we first revisit the three conditions and then empirically validate the sparsity of interactions.

## B.1 COMMON CONDITIONS FOR SPARSE INTERACTIONS

**Condition 1.** *The DNN does not encode extremely high-order interactions:* $\forall\, T \in \{T \subseteq N || T| \geq M+1\},\ I_T^{\text{and}} = 0$.

Condition 1 is common because extremely high-order interactions usually represent very complex and over-fitted patterns, which are unlikely to be learned by a well-trained DNN in real scenarios.

**Condition 2.** *Let* $\bar{u}^{(k)} \overset{\text{def}}{=} \mathbb{E}_{|S|=k}[v(\boldsymbol{x}_S) - v(\boldsymbol{x}_\emptyset)]$ *denote the average classification confidence of the DNN over all masked samples* $\boldsymbol{x}_S$ *with* $k$ *unmasked input variables. This average classification confidence monotonically increases when* $k$ *increases:* $\forall\, k' \leq k,\ \bar{u}^{(k')} \leq \bar{u}^{(k)}$.

Condition 2 implies that a well-trained DNN is likely to have higher average classification confidence for less masked input samples.

**Condition 3.** *Given the average classification confidence* $\bar{u}^{(k)}$ *of samples with* $k$ *unmasked input variables, there is a polynomial lower bound for the average classification confidence with* $k'(k' \leq k)$ *unmasked input variables:* $\forall\, k' \leq k,\ \bar{u}^{(k')} \geq (\frac{k'}{k})^p\, \bar{u}^{(k)}$, *where* $p > 0$ *is a constant.*

Condition 3 suggests that the classification confidence of the DNN remains relatively stable even when presented with masked input samples. In real-world applications, the classification or detection of masked or occluded samples frequently occurs. As a result, a well-trained DNN typically develops the ability to classify such masked inputs by leveraging local information, which can be derived from the visible portions of the input. Consequently, the model should not produce a substantially reduced confidence score for masked samples.

## B.2 EMPIRICAL VERIFICATION OF INTERACTION SPARSITY ON LLMS

Given an input sample, the preceding subsection presents sufficient conditions for interaction sparsity. In this subsection, we further provide empirical evidence showing that the interactions extracted from large language models are sparse in practice.

Specifically, we randomly select 20 input samples and extract the interactions encoded by the corresponding LLMs for these samples. Following (Ren et al., 2024a), we normalize the interaction strengths by the maximum absolute interaction strength within each model-dataset setting, and sort all interactions in descending order according to their absolute values. We aggregate the interactions from all selected samples into a single pool and display them together in one figure for each model-dataset setting.

We conduct experiments on multiple LLMs and datasets, including Llama2-7B-Chat and Qwen2.5-7B-Instruct on the GoEmotions dataset, Qwen2.5-3B-Instruct on the Unilaw-R1-Data dataset, and Gemma-3-4B-it on the Databricks-Dolly-15k dataset. For each model-dataset setting, we compute the interaction strengths for the selected samples, merge all interactions across these samples, and analyze their sorted distributions.

As shown in Figure 6, only a small number of interactions have relatively large magnitudes, while the vast majority of interactions are close to zero. This phenomenon consistently holds across different LLMs and datasets, providing strong empirical evidence that the interactions are highly sparse in practice.

# C COMPUTATIONAL COST AND PRACTICAL SCALABILITY

Similar to the computation of Shapley values (Shapley et al., 1953), interaction extraction incurs exponential computational cost in the worst case, as it requires evaluating masked states over the

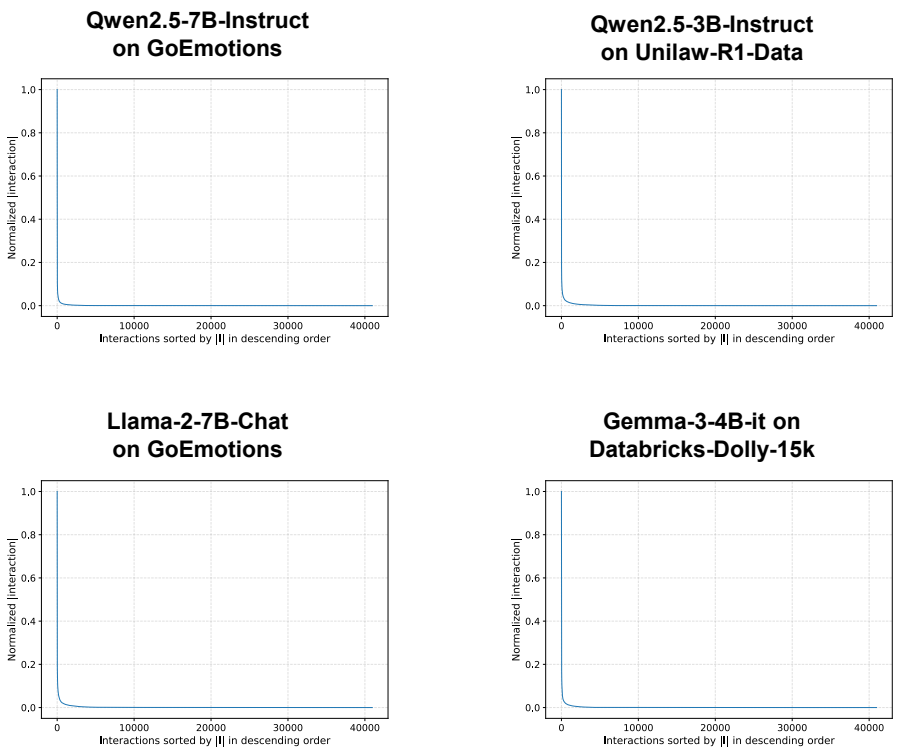


Figure 6: Empirical verification of interaction sparsity on LLMs. We aggregate the interactions extracted from all selected samples into a single pool, normalize their strengths, and sort them in descending order according to their absolute values. The results show that only a small fraction of interactions have relatively large magnitudes, while the vast majority are close to zero. These results provide strong empirical evidence that the interactions are highly sparse in practice.

input. Nevertheless, prior work has shown that this cost can be substantially mitigated in practice by exploiting the sparsity of interactions and adopting efficient approximation algorithms.

For example, the Sparse Möbius Transform reduces the complexity to $O(nK \log n)$ when only $K$ salient interactions exist (Kang et al., 2024). Spectral Explainer efficiently identifies important interactions for LLMs with long input sequences (Kang et al., 2025), while ProxySPEX further improves scalability by using lightweight proxy models to approximate the local decision behavior of the original model (Butler et al., 2025). Together with practical engineering techniques (Li & Zhang, 2023a), these methods provide a practical pathway toward scalable interaction-based explanations.

In addition, for LLMs, interactions do not necessarily need to be extracted at the level of individual tokens. Depending on the analysis objective, one can compute interactions among phrases, words, or even sentences, which effectively reduces the number of input variables and substantially alleviates the sequence-length issue. Moreover, analyzing the evolution of interactions during the SFT process typically requires estimating interaction distributions rather than exhaustively enumerating all possible interactions. In such cases, selectively computing interactions over a small set of salient input variables is often sufficient to obtain a stable interaction distribution for generalization analysis. Consequently, the computational cost can be handled efficiently in typical experimental settings.

Therefore, although exact interaction extraction remains computationally expensive in theory, existing approximation algorithms, flexible choices of interaction granularity, and practical engineering techniques make interaction-based explanations increasingly feasible for modern large-scale models. Interaction-based explanations have also been applied to diverse DNNs, including LLMs, quantitative trading models, and pedestrian detection models in autonomous driving, as well as industrial model-evaluation scenarios.

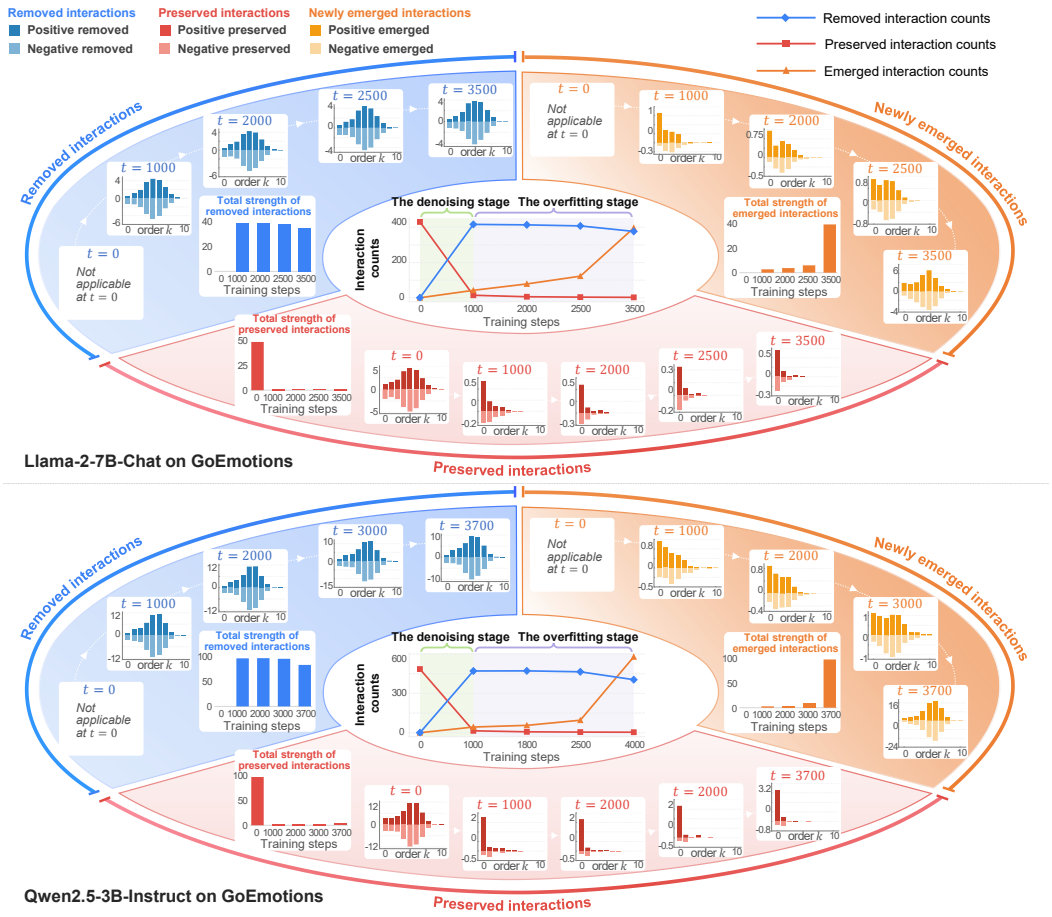


Figure 7: Additional results on the evolution of newly emerged, removed, and preserved interactions during SFT. The figure shows Llama-2-7B-Chat and Qwen2.5-3B-Instruct on GoEmotions. Additional results are presented on the next pages.

# D More Experimental Results

## D.1 Additional Results on the Evolution of Interactions

In this appendix, we provide additional results on the evolution of interactions during SFT across more LLMs and datasets. Figure 7 presents results for Llama-2-7B-Chat and Qwen2.5-3B-Instruct on GoEmotions, Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct on Unilaw-R1-Data, and Gemma-3-4B-it on Databricks-Dolly-15k. These results are consistent with the observations in the main paper. In the early denoising stage, LLMs rapidly remove many mutually canceling interactions while preserving only a small number of low-order interactions. In the later overfitting stage, many high-order mutually canceling interactions gradually emerge again.

## D.2 Additional Results on the Representation Quality of Interactions

In this appendix, we provide additional results on the representation quality of newly emerged, removed, and preserved interactions during SFT. Figure 8 reports results for Qwen2.5-3B-Instruct on GoEmotions, Qwen2.5-3B-Instruct on Unilaw-R1-Data, and Gemma-3-4B-it on Databricks-Dolly-15k. These results are consistent with the observations in the main paper. Newly emerged interactions exhibit high representation quality only in the short denoising stage, but become less generalizable and more mutually canceling in the later stage. Removed interactions remain low-quality throughout SFT, while preserved interactions improve after noisy interactions are removed. Since only one LLM is fine-tuned on Databricks-Dolly-15k, no baseline LLM is available for measuring interaction

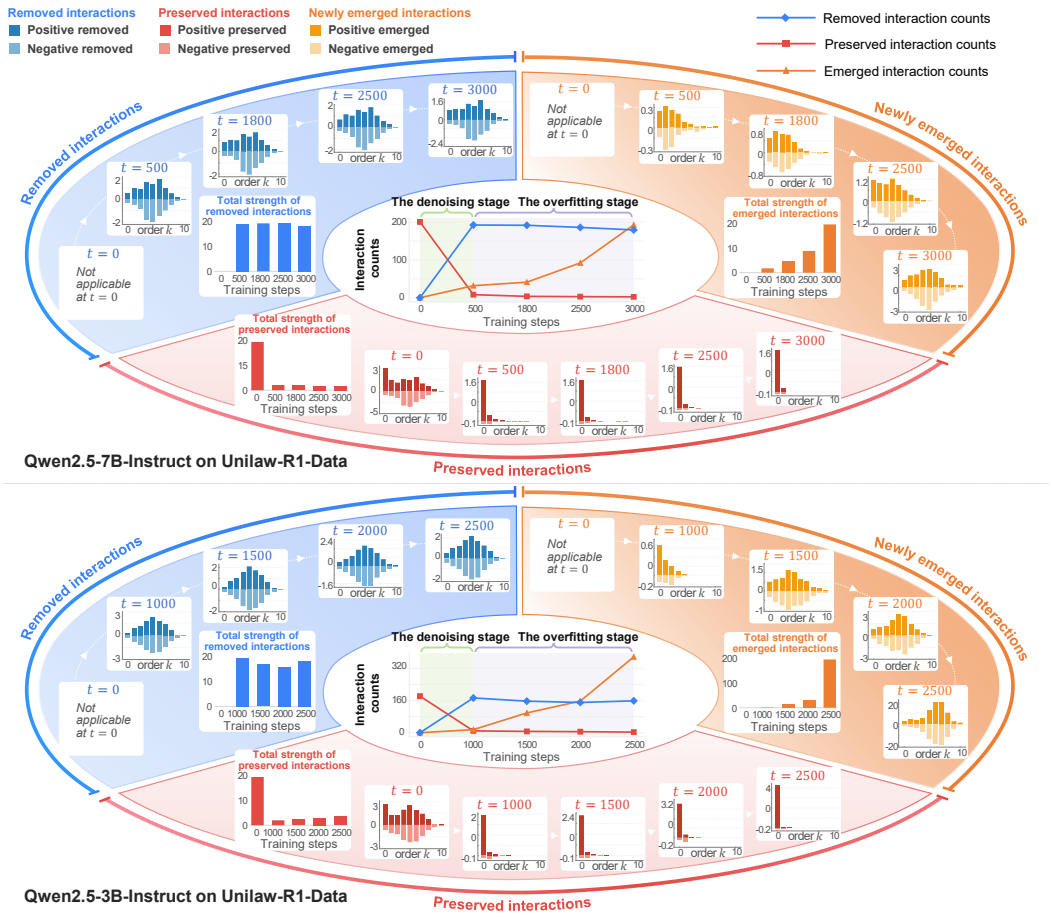


Figure 7: Additional results on the evolution of newly emerged, removed, and preserved interactions during SFT. The figure shows Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct on Unilaw-R1-Data. Additional results are presented on the next page.

generalizability $\gamma$ on this dataset. Therefore, we only report the uncancelled-effect ratio $\rho$ for Gemma-3-4B-it.

### D.3 Downstream task performance throughout SFT

The preceding interaction analyses identify two distinct stages during SFT: a short first stage dominated by the removal of noise-like interactions, followed by a longer second stage in which many newly emerged interactions gradually accumulate. To examine whether these interaction-level changes are reflected in downstream task performance, we evaluate different LLM checkpoints before SFT, during the first stage, and during the second stage.

We use standard task-specific evaluation metrics. GoEmotions is a multi-label emotion classification dataset, for which we report Accuracy, Micro-F1, and Macro-F1. Unilaw-R1-Data and Databricks-Dolly-15k require free-form text generation, for which label-based classification metrics are not directly applicable. Therefore, we report ROUGE-L F1, which measures the longest-common-subsequence overlap between the generated response and the reference response. Higher values indicate better performance for all reported metrics.

Table 1 reports the results on GoEmotions. Both LLMs exhibit substantial performance improvements during the first stage. For Qwen2.5-7B-Instruct, Accuracy increases from 0.130 before SFT to 0.580 at step 1000, while Micro-F1 increases from 0.267 to 0.654 and Macro-F1 increases from 0.204 to 0.416. All three metrics reach their highest values at this early checkpoint. During the second stage,

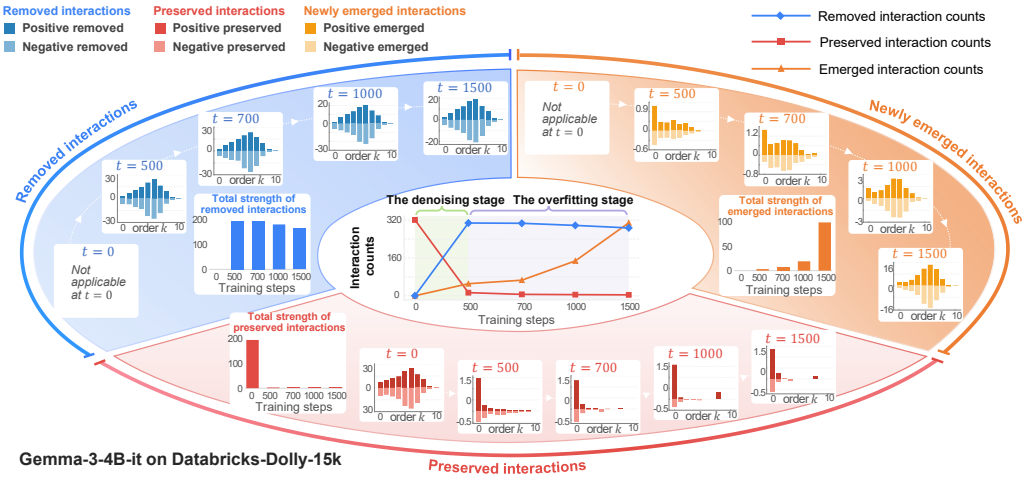


Figure 7: Additional results on the evolution of newly emerged, removed, and preserved interactions during SFT. The figure shows Gemma-3-4B-it on Databricks-Dolly-15k. Across these LLMs and datasets, the results consistently show that SFT first removes many mutually canceling interactions in a short denoising stage, while many high-order mutually canceling interactions emerge again in the later overfitting stage.

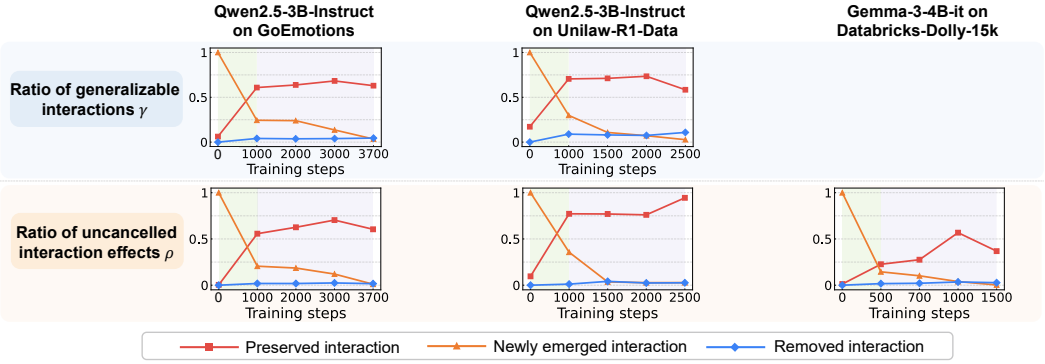


Figure 8: Additional results on the representation quality of newly emerged, removed, and preserved interactions during SFT. The figure shows Qwen2.5-3B-Instruct on GoEmotions, Qwen2.5-3B-Instruct on Unilaw-R1-Data, and Gemma-3-4B-it on Databricks-Dolly-15k. Since no baseline LLM is trained on Databricks-Dolly-15k, interaction generalizability $\gamma$ is not evaluated for Gemma-3-4B-it, and only the uncancelled-effect ratio $\rho$ is reported.

none of the later checkpoints exceeds the performance obtained at step 1000. At step 4000, Accuracy, Micro-F1, and Macro-F1 decrease to 0.530, 0.593, and 0.339, respectively.

Llama-3-8B-Instruct exhibits the same overall trend. At step 1000, Accuracy increases from 0.120 to 0.570, Micro-F1 increases from 0.260 to 0.639, and Macro-F1 increases from 0.199 to 0.381. Continued SFT does not produce further gains. At step 3000, the three metrics decrease to 0.500, 0.573, and 0.366, respectively. Although the later trajectories are not strictly monotonic for every metric, all later checkpoints remain below the best result achieved during the first stage.

Table 2 presents the results on the two free-form generation datasets. On Unilaw-R1-Data, ROUGE-L F1 increases from 0.157 before SFT to 0.185 at step 1000. Subsequent checkpoints yield values between 0.181 and 0.184, indicating that continued SFT does not improve upon the best result obtained during the first stage.

On Databricks-Dolly-15k, ROUGE-L F1 increases substantially from 0.232 before SFT to 0.367 at step 300. During the second stage, performance fluctuates between 0.347 and 0.366, but none of the

Table 1: Downstream performance on GoEmotions throughout SFT. The first stage corresponds to the short denoising stage identified by the interaction analysis, while the second stage denotes subsequent SFT. Bold numbers indicate the best result for each model and metric.

| Model | Step | Stage | Accuracy ↑ | Micro-F1 ↑ | Macro-F1 ↑ |
|---|---|---|---|---|---|
| Qwen2.5-7B-Instruct | 0 | Before SFT | 0.130 | 0.267 | 0.204 |
| Qwen2.5-7B-Instruct | 1000 | First stage | **0.580** | **0.654** | **0.416** |
| Qwen2.5-7B-Instruct | 1800 | Second stage | 0.570 | 0.645 | 0.407 |
| Qwen2.5-7B-Instruct | 2500 | Second stage | 0.520 | 0.581 | 0.365 |
| Qwen2.5-7B-Instruct | 4000 | Second stage | 0.530 | 0.593 | 0.339 |
| Llama-3-8B-Instruct | 0 | Before SFT | 0.120 | 0.260 | 0.199 |
| Llama-3-8B-Instruct | 1000 | First stage | **0.570** | **0.639** | **0.381** |
| Llama-3-8B-Instruct | 1800 | Second stage | 0.560 | 0.630 | 0.371 |
| Llama-3-8B-Instruct | 2500 | Second stage | 0.530 | 0.593 | 0.373 |
| Llama-3-8B-Instruct | 3000 | Second stage | 0.500 | 0.573 | 0.366 |

Table 2: Downstream performance on the free-form generation datasets throughout SFT. The first stage corresponds to the short denoising stage, while the second stage denotes subsequent SFT. Bold numbers indicate the best result within each model and dataset setting.

| Dataset | Model | Step | Stage | ROUGE-L F1 ↑ |
|---|---|---|---|---|
| Unilaw-R1-Data | Qwen2.5-3B-Instruct | 0 | Before SFT | 0.157 |
| Unilaw-R1-Data | Qwen2.5-3B-Instruct | 1000 | First stage | **0.185** |
| Unilaw-R1-Data | Qwen2.5-3B-Instruct | 1500 | Second stage | 0.183 |
| Unilaw-R1-Data | Qwen2.5-3B-Instruct | 1800 | Second stage | 0.184 |
| Unilaw-R1-Data | Qwen2.5-3B-Instruct | 2500 | Second stage | 0.181 |
| Databricks-Dolly-15k | Gemma-3-4B-it | 0 | Before SFT | 0.232 |
| Databricks-Dolly-15k | Gemma-3-4B-it | 300 | First stage | **0.367** |
| Databricks-Dolly-15k | Gemma-3-4B-it | 500 | Second stage | 0.354 |
| Databricks-Dolly-15k | Gemma-3-4B-it | 1000 | Second stage | 0.366 |
| Databricks-Dolly-15k | Gemma-3-4B-it | 1300 | Second stage | 0.347 |

later checkpoints surpasses the early best result. In particular, ROUGE-L F1 decreases to 0.347 at the final evaluated checkpoint.

Overall, the best downstream performance is reached during the first stage in all four reported model and dataset settings. This stage is also characterized by the rapid removal of many noise-like interactions. During the second stage, newly emerged interactions continue to accumulate, whereas downstream performance no longer improves and generally declines relative to the early best checkpoint. These results demonstrate a consistent temporal correspondence between interaction-level dynamics and task-level performance. Thus, the downstream results provide additional evidence that the short denoising stage captures the primary beneficial effects of SFT, while prolonged fine-tuning offers limited additional gains and may gradually impair generalization.

### D.4 Examples of AND-OR Logical Models Explaining LLMs

In this section, to demonstrate that AND-OR logical models can faithfully explain large language models, we present examples showing that, given specific input prompts, AND-OR logical models can be constructed to faithfully explain both the DeepSeek-r1-distill-llama-8B (Guo et al., 2025) and Qwen2.5-7B (Bai et al., 2023) models, hereafter referred to as the DeepSeek and Qwen models, respectively. Figure 9 illustrates the AND-OR logical models used to explain DeepSeek and Qwen under different input prompts.

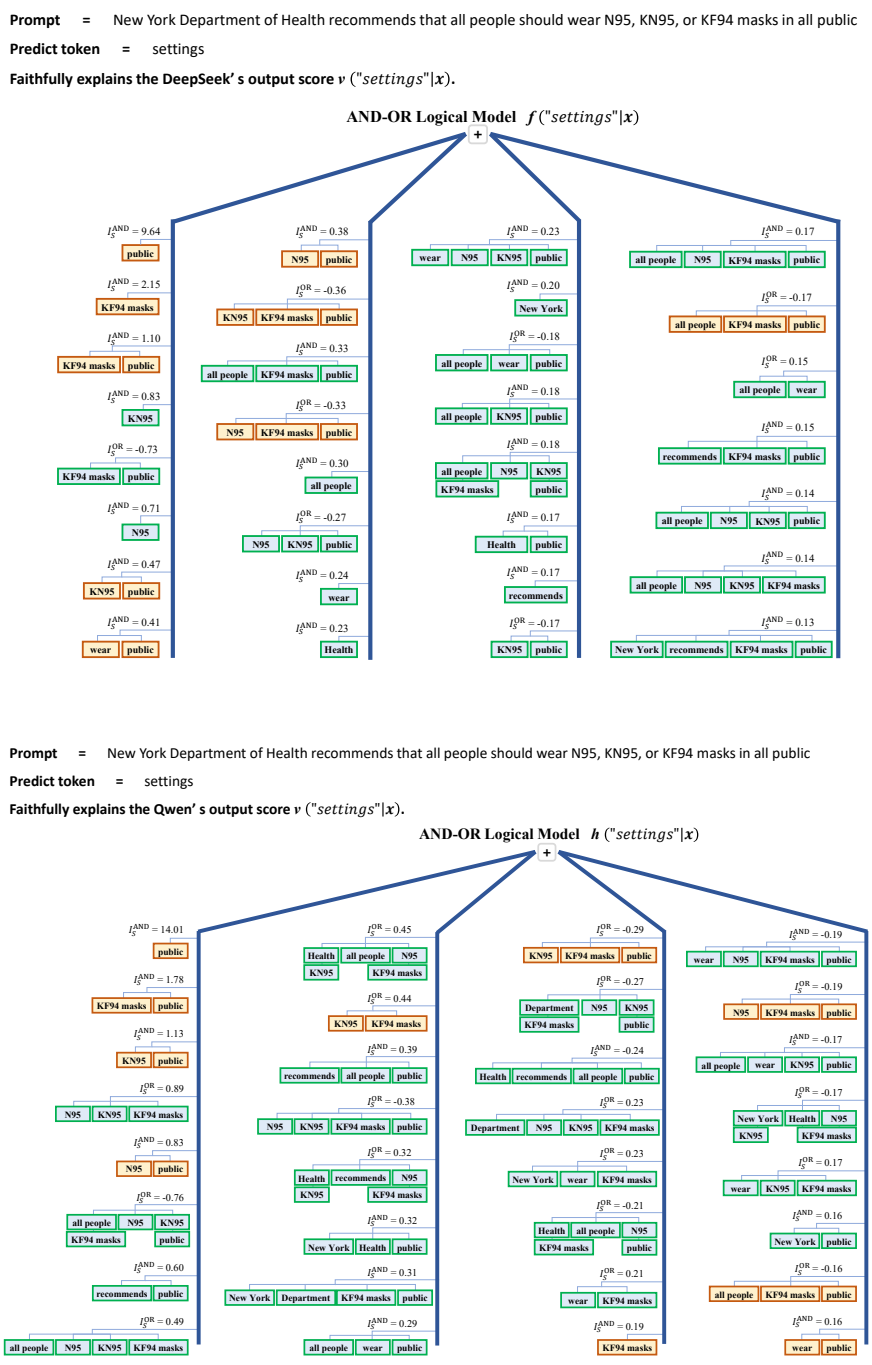


Figure 9: An example of AND-OR logical models constructed to faithfully explain the output scores of the DeepSeek model (top) and the Qwen model (bottom) on a single sample. A further example is presented on the next page.

# E Experimental Details

## E.1 Training Settings

**Models and datasets.** We conduct supervised fine-tuning (SFT) experiments on multiple LLMs across different datasets. For the GoEmotions dataset (Demszky et al., 2020), we fine-tune Qwen2.5-3B-Instruct (Team, 2024), Qwen2.5-7B-Instruct (Team, 2024), Llama-2-7B-Chat (Touvron et al.,

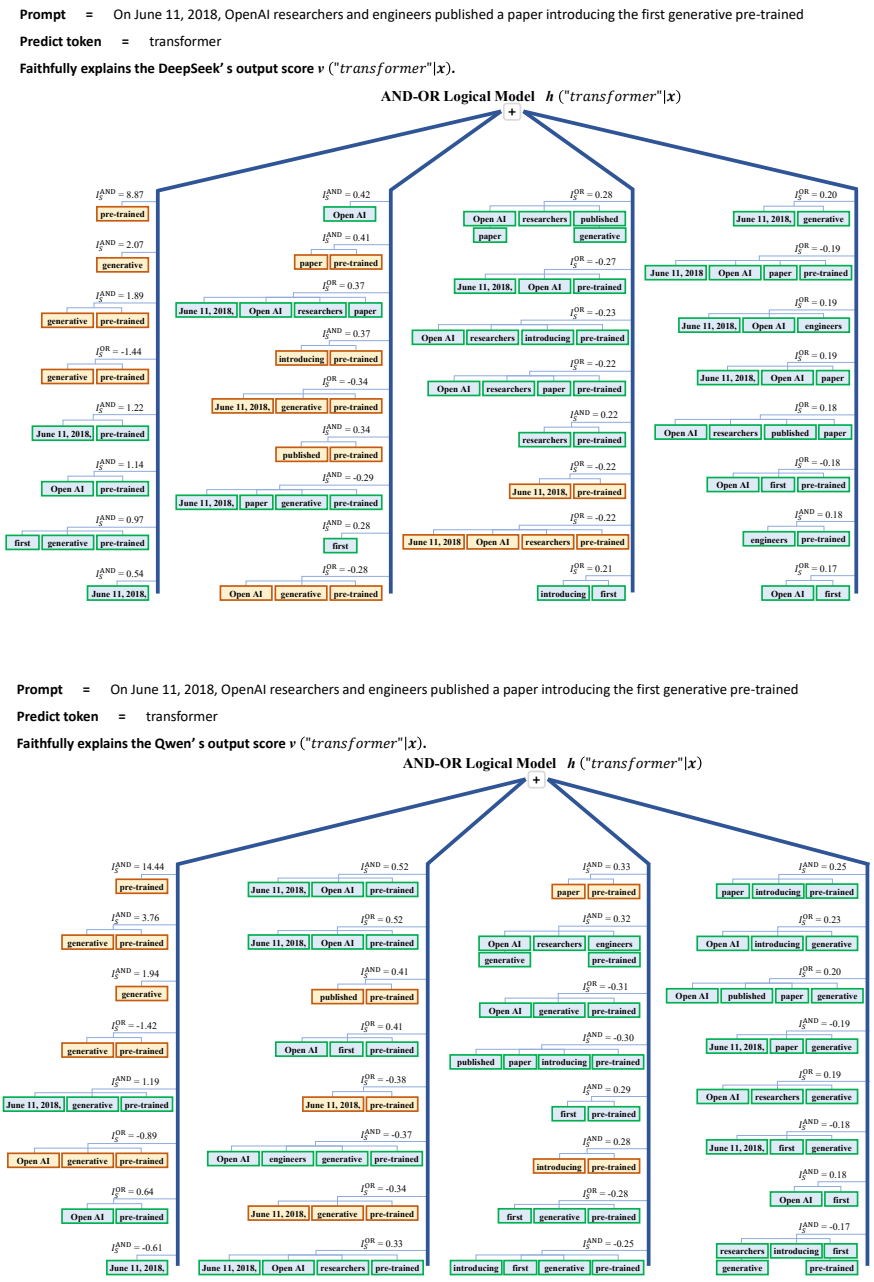


Figure 9: Another example of AND-OR logical models explaining the DeepSeek model (top) and the Qwen model (bottom) on a different sample.

2023), and Llama-3-8B-Instruct (Grattafiori et al., 2024). For the Unilaw-R1-Data dataset (Cai et al., 2025), we fine-tune Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. For the Databricks-Dolly-15k dataset (Conover et al., 2023), we fine-tune Gemma-3-4B-it (Team et al., 2025). All datasets are formatted into instruction-response style examples, and the corresponding model-specific chat template is used for each model.

*GoEmotions.* GoEmotions (Demszky et al., 2020) is a large-scale English emotion classification dataset collected from Reddit comments. It contains approximately 58K carefully curated comments annotated with 27 fine-grained emotion categories and a neutral label. Each instance consists of a text comment and one or more emotion labels, making the dataset suitable for both multi-class and

multi-label emotion classification. The dataset is publicly available, and its hosting GitHub repository is released under the Apache License 2.0.

*Unilaw-R1-Data.* Unilaw-R1-Data (Cai et al., 2025) is a legal-domain instruction dataset constructed for training legal reasoning models. The dataset is derived from objective question-answering entries in the legal domain, including JEC-QA and the authors' primary legal reasoning data, and is further distilled and partitioned into supervised fine-tuning and reinforcement learning subsets. The dataset is publicly available, but the original source does not specify an explicit dataset license.

*Databricks-Dolly-15k.* Databricks-Dolly-15k (Conover et al., 2023) is an open-source English instruction-following dataset consisting of more than 15K human-generated instruction-response records. The data were created by Databricks employees and cover multiple instruction categories, including brainstorming, classification, closed-form question answering, generation, information extraction, open-domain question answering, and summarization. The dataset is publicly available under the CC BY-SA 3.0 license.

**Fine-tuning details.** All SFT experiments are implemented with LLaMA-Factory (Zheng et al., 2024). We adopt LoRA (Hu et al., 2021) as the parameter-efficient fine-tuning method. Unless otherwise specified, the LoRA rank is set to $r = 8$, and LoRA adapters are applied to all eligible linear modules. The maximum sequence length is set to 8192, and the maximum number of training samples is capped at 100,000 when applicable. During training, the loss is computed only on the labeled response tokens, while prompt tokens are excluded from the loss.

We use a peak learning rate of $1 \times 10^{-4}$ with a cosine learning-rate scheduler and a warmup ratio of 0.1. Models are trained with bfloat16 precision for 30 epochs. The per-device training batch size is set to 1, and gradients are accumulated over 8 steps. All experiments are conducted on 8 NVIDIA Tesla V100-PCIE-32GB GPUs, resulting in an effective batch size of 64 samples per optimization step.

### E.2 Baseline LLMs for Measuring Interaction Generalizability.

To quantify the generalizability of interactions, we introduce a baseline LLM $v'$ for each target LLM $v$, where $v'$ is fine-tuned on the same dataset but has a different architecture from $v$. For the GoEmotions dataset, we use the fine-tuned Llama-2-7B-Chat model as the baseline LLM for the Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Llama-3-8B-Instruct target models. When the target model is Llama-2-7B-Chat, we instead use the fine-tuned Qwen2.5-3B-Instruct model as the baseline LLM. For the Unilaw-R1-Data dataset, since we fine-tune two LLMs, Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, we use them as the baseline LLMs for each other. For the Databricks-Dolly-15k dataset, we fine-tune only one LLM, Gemma-3-4B-it, and therefore do not evaluate the generalizability of interactions on this dataset.

When plotting the trajectories of $\gamma$ and $\rho$, newly emerged interactions and removed interactions are not available before SFT because they are defined by their changes along the SFT trajectory. For consistency of visualization, we use conventional boundary values for such undefined entries, setting $\gamma = \rho = 1$ for newly emerged interactions and $\gamma = \rho = 0$ for removed interactions. These boundary values only serve to anchor the curves and do not affect our quantitative comparisons.

### E.3 Computer Resources

All LLMs can be trained on 8 NVIDIA Tesla V100-PCIE-32GB GPUs, and can be finished within 120 hours. Calculating all interactions for an input sample takes an average of 30-50 seconds.

### E.4 Details About How to Calculate Interactions for Different LLMs

Interaction-based explanations have been widely applied to real-world DNNs, largely due to prior efforts (Ren et al., 2024b; Li & Zhang, 2023b) that substantially reduce the computational cost of interaction extraction. Moreover, analyses of the evolution of interactions during the SFT of LLMs typically require computing only interaction distributions rather than exhaustively enumerating all interactions. As a result, selectively computing interactions over a small number of salient input variables is usually sufficient to obtain stable interaction distributions for analysis. Consequently, the computation can be handled efficiently in typical experimental settings.

Extensive experiments on large language models further support this observation. We find that LLMs usually attend to only a small number of salient text segments when making predictions. As a result, explanations based on fewer than 10 salient text segments are often sufficient to faithfully capture the LLM's inference patterns. This empirical property significantly improves the computational efficiency of interaction-based explanations and ensures their practical applicability to large-scale models and datasets.

Specifically, we followed (Chen et al., 2024) to define each input variable as the embedding vector(s) of a word and calculate interactions. Besides, the distribution of interactions is computed by averaging the distribution across different samples. Furthermore, we followed (Chen et al., 2024) to use the first target token of the response for interaction analysis.

**Algorithm 1** Compute AND and OR Interactions and Select Salient Ones

**Require:** Deep neural network $v$, input sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$, set of indices $N = \{1, 2, \ldots, n\}$, small noise threshold $\zeta$, significance threshold $\tau$, convergence threshold $\epsilon$.
**Ensure:** AND interactions $I_T^{\text{and}}$, OR interactions $I_T^{\text{or}}$, and interaction sets $\Omega^{\text{and}}$ and $\Omega^{\text{or}}$.

Initialize learnable parameters $\{\gamma_L\}$ and $\{\delta_L\}$ for all $L \subseteq N$.
Compute baseline output $v(\boldsymbol{x}_\emptyset)$, where $\boldsymbol{x}_\emptyset$ is the masked sample with all variables removed.
Initialize previous loss $\mathcal{L}_{\text{prev}} \leftarrow \infty$.
**repeat**
  **for** each subset $L \subseteq N$ **do**
    Compute masked sample $\boldsymbol{x}_L$ by removing variables not in $L$.
    Compute network output $v(\boldsymbol{x}_L)$.
    Compute noise term $\delta_L$ constrained in $[-\zeta, \zeta]$, where $\zeta = 0.01 \cdot |v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)|$.
    Decompose $v(\boldsymbol{x}_L)$ into AND and OR components:
    $u_L^{\text{and}} \leftarrow 0.5 \cdot (v(\boldsymbol{x}_L) - \delta_L) + \gamma_L$
    $u_L^{\text{or}} \leftarrow 0.5 \cdot (v(\boldsymbol{x}_L) - \delta_L) - \gamma_L$
  **end for**
  **for** each subset $T \subseteq N$ **do**
    Compute AND interaction $I_T^{\text{and}}$:

$$I_T^{\text{and}} \leftarrow \sum_{L \subseteq T} (-1)^{|T|-|L|} u_L^{\text{and}}$$

    Compute OR interaction $I_T^{\text{or}}$:

$$I_T^{\text{or}} \leftarrow -\sum_{L \subseteq T} (-1)^{|T|-|L|} u_{N \setminus L}^{\text{or}}$$

  **end for**
  Compute current loss $\mathcal{L} \leftarrow \sum_{T \subseteq N} \left(|I_T^{\text{and}}| + |I_T^{\text{or}}|\right)$.
  Optimize parameters $\{\gamma_L\}$ and $\{\delta_L\}$ to minimize $\mathcal{L}$.
  Check for convergence: $|\mathcal{L} - \mathcal{L}_{\text{prev}}| < \epsilon$.
  Update previous loss: $\mathcal{L}_{\text{prev}} \leftarrow \mathcal{L}$.
**until** convergence
Select AND interactions:

$$\Omega^{\text{and}} \leftarrow \{T \subseteq N : |I_T^{\text{and}}| > \tau\}$$

Select OR interactions:

$$\Omega^{\text{or}} \leftarrow \{T \subseteq N : |I_T^{\text{or}}| > \tau\}$$

**return** $I_T^{\text{and}}$, $I_T^{\text{or}}$, $\Omega^{\text{and}}$, and $\Omega^{\text{or}}$.

## F Details to Extract the Sparsest AND-OR Interactions

A method is proposed (Li & Zhang, 2023a; Chen et al., 2024) to simultaneously extract AND interactions $I_T^{\text{and}}$ and OR interactions $I_T^{\text{or}}$ from the network output. Given a masked sample $\boldsymbol{x}_L$, Li & Zhang (2023a) proposed to learn a decomposition $v(\boldsymbol{x}_L) = u_L^{\text{and}} + u_L^{\text{or}}$ towards the sparsest

interactions. The component $u_L^{\text{and}}$ was explained by AND interactions, and the component $u_L^{\text{or}}$ was explained by OR interactions. Specifically, they decomposed $v(\boldsymbol{x}_L)$ into $u_L^{\text{and}} = 0.5 \cdot v(\boldsymbol{x}_L) + \gamma_L$ and $u_L^{\text{or}} = 0.5 \cdot v(\boldsymbol{x}_L) - \gamma_L$, where $\{\gamma_L : L \subseteq N\}$ is a set of learnable variables that determine the decomposition. In this way, the AND interactions and OR interactions can be computed, *i.e.*, $I_T^{\text{and}} = \sum_{L \subseteq T} (-1)^{|T|-|L|} u_L^{\text{and}}$, and $I_T^{\text{or}} = -\sum_{L \subseteq T} (-1)^{|T|-|L|} v_{N \setminus L}^{\text{or}}$.

The parameters $\{\gamma_L\}$ were learned by minimizing the following LASSO-like loss to obtain sparse interactions:

$$\min_{\{\gamma_L\}} \sum_{T \subseteq N} |I_T^{\text{and}}| + |I_T^{\text{or}}| \tag{15}$$

**Removing small noises.** A small noise $\delta$ in the network output may significantly affect the extracted interactions, especially for high-order interactions. Thus, (Li & Zhang, 2023a) proposed to learn to remove a small noise term $\delta_T$ from the computation of AND-OR interactions. Specifically, the decomposition was rewritten as $u_L^{\text{and}} = 0.5(v(\boldsymbol{x}_L) - \delta_L) + \gamma_L$ and $u_L^{\text{or}} = 0.5(v(\boldsymbol{x}_L) - \delta_L) - \gamma_L$. Thus, the parameters $\{\delta_L\}$ and $\{\gamma_L\}$ are simultaneously learned by minimizing the loss function in Eq. (15). The values of $\{\delta_L\}$ were constrained in $[-\zeta, \zeta]$ where $\zeta = 0.01 \cdot |v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)|$. Therefore, we can also set $\epsilon = \zeta$ to ensure the fidelity of using the logical model function $\phi(\cdot)$ to mimic the DNN function $v(\cdot)$.

**Algorithm of extracting AND-OR interactions.** The technical details of computing $I_T^{\text{and}}$ and $I_T^{\text{or}}$ are provided in the following pseudocode in Algorithm 1.

## G PRACTICAL IMPLICATIONS FOR MONITORING SFT

Our analysis suggests a potential use of interactions for monitoring SFT. At each evaluated checkpoint, one can measure the proportion of generalizable interactions, $\gamma_t = \gamma(\Omega_t^{\text{and}}, \Omega_t^{\text{or}})$, on a fixed set of analysis samples, and examine its trajectory alongside the validation loss. A sustained decline in $\gamma_t$, particularly when the validation loss has stopped improving, may indicate that continued SFT is introducing increasingly model-specific interactions. This combination could serve as a signal to consider early stopping.

A decline in $\gamma_t$ alone should not determine the stopping point. If the validation loss continues to improve, the two signals should be examined together before ending training. A substantial decline in $\gamma_t$ may also motivate a closer examination of the training data and evaluation on additional held-out samples. Improving the data, for example through data augmentation, could then be explored in a subsequent training run. Establishing a stopping threshold and evaluating whether this strategy improves downstream generalization remain directions for future work.