Can Input Attributions Interpret the Inductive Reasoning Process in In-Context Learning?

Anonymous ACL submission

Abstract

Interpreting the internal process of neural models has long been a challenge. This challenge remains relevant in the era of large language models (LLMs) and in-context learning (ICL); for example, ICL poses a new issue of interpreting which example in the few-shot examples contributed to identifying/solving the task. To this end, in this paper, we design synthetic diagnostic tasks of inductive reasoning, inspired by the generalization tests in linguistics; here, most in-context examples are ambiguous w.r.t. their underlying rule, and one critical example disambiguates the task demonstrated. The question is whether conventional input attribution (IA) methods can track such a reasoning process, i.e., identify the influential example, in ICL. Our experiments provide several practical findings; for example, a certain simple IA method works the best, and the larger the model, the generally harder it is to interpret the ICL with gradient-based IA methods.[1]

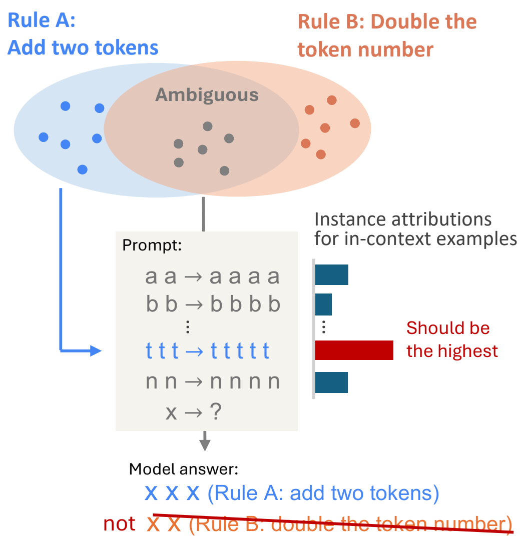

Figure 1: Overview of our experimental setup. The majority of in-context examples (gray) are ambiguous, supporting either Rule A of adding two tokens or Rule B of doubling tokens. A single disambiguating example (blue) reveals that Rule A is correct. We investigate whether input attribution (IA) methods can track such an inductive reasoning process.

1 Introduction

In past years, input attribution (IA) methods, e.g., gradient norm (Simonyan et al., 2014; Li et al., 2016a), have typically been employed in the natural language processing (NLP) field to interpret input–output associations exploited by neural NLP models (Vinyals and Le, 2015; Li et al., 2016b). Recently, large language models (LLMs) and mechanistic interpretability (MI) research (Olah et al., 2020; Bereska and Gavves, 2024) have shifted the research focus to understanding the *circuits* within LLMs by intervening in their internal representations. Despite the enriched scope of research, such rapid progress has missed some intriguing questions bridging the IA and MI eras: in particular, *do conventional IA methods still empirically work in the modern NLP setting, specifically in the context of LLMs and in-context learning (ICL)?*

In this paper, we revisit IA methods in interpreting LLM-based ICL (Brown et al., 2020). Specifically, we assess how well IA methods can track the most influential example in a few-shot examples. This question is worth investigating for several reasons. First, input attribution would still serve as a necessary and sufficient explanation in typical practical cases; some users might simply seek which part of the context is heavily referred to by an LLM system rather than LLMs' internal processes identified by MI methods. Second, the modern NLP setting, specifically ICL, differs from the conventional settings where IA methods have been tested — identifying the input-output association within a specific test instance

[1] We will make our data and scripts public upon the publication of this paper.

(X_k, y_k). In contrast, applying IA methods to the entire ICL input (few-shot examples) already entails tracking the *learning* process as well as the input-output association within a specific target instance. That is, this extended scope includes the interpretation of which *example* among the demonstrations $[(X_1, y_1), \cdots, (X_{k-1}, y_{k-1})]$ contributed to identifying the targeted task/rule and then answer a target question X_k. This is rather a type of instance-based interpretation of neural models (Wachter et al., 2017; Charpiat et al., 2019; Hanawa et al., 2021), and it has been little explored such interpretation is feasible with IA methods.

To test IA methods in ICL, we introduce a test suite comprising controlled synthetic inductive reasoning tasks. Otherwise, formally defining such informativeness and assessing IA methods is challenging, especially in a wild, natural setting; critical examples may not be unique (Min et al., 2022), a gap might exist between faithfulness and plausibility perspectives (Bastings et al., 2022), and a model can rather rely on prior knowledge without using any input examples (Liu et al., 2022). Our task design, inspired by the *poverty of the stimulus scenario* (Wilson, 2006; Perfors et al., 2011; McCoy et al., 2018, 2020; Yedetore et al., 2023) or *mixed signals generalization test* (Warstadt and Bowman, 2020; Mueller et al., 2024), introduces one inherently unique *aha* example in input demonstrations. This *aha* example, when paired with any of the other examples, triggers the identification of the underlying reasoning rule. More specifically, most in-context examples are *ambiguous* in the sense that they are compatible with several rules (e.g., adding two tokens or doubling the number of tokens, in the case of Figure 1), and only one *disambiguating* (*aha*) example resolves the ambiguity and limits the correct rule to be unique (ttt→ttttt disambiguates the rule to be *adding* one in Figure 1). The question is whether such an informative example can be empirically caught by IA methods.

Our experiments reveal several findings:

- Gradient norm, the simplest IA method, frequently outperforms other interpretability methods (e.g., integrated gradient), suggesting that the advantage of more recently proposed IA methods does not always generalize in interpreting ICL×LLM.
- Our tested interpretability methods, including simple gradient norm, did not work stably across different tasks and models, posing their general limitations in interpreting ICL with IA methods.
- Different interpretability methods exhibited different properties with respect to scaling the number of in-context examples or model size; for example, IA methods perform better in many-shot scenarios, whereas a particular baseline interpretability method (i.e., self-answer) works well on larger models.

2 Preliminary

2.1 Input attribution (IA) methods

Input attribution (IA) methods are commonly-used techniques for interpreting and explaining the predictions of machine learning models (Denil et al., 2014; Li et al., 2016a; Poerner et al., 2018; Arras et al., 2019, etc.). Specifically, IA methods determine how much each input feature contributes to a particular prediction; that is, given input tokens $X := [x_1, \ldots, x_n]$ and output y, the IA methods yield the strength of contribution $S(x_i)$ of each input x_i to the output y. Note that the input X in ICL consists of several in-context examples (§ 2.2), and the answer to the target question is denoted as y. We examine the following four representative IA methods in the ICL×LLM context:

Input erasure (IE) IE (Li et al., 2016c) measures how impactful erasing a certain token x_i from the input prompt is with respect to outputing y_t:

$$S_{\mathrm{IE}}(x_i, y_t; X) = q(y_t|\boldsymbol{X}) - q(y_t|\boldsymbol{X}_{\neg i}), \quad (1)$$

where $\boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ denotes the sequence of input token embeddings, with each $\boldsymbol{x}_i \in \mathbb{R}^d$ being a d-dimensional vector corresponding to the i-th token in the input. $\boldsymbol{X}_{\neg i} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n]$ denotes the sequence of input token embeddings without $\boldsymbol{x}_i$. We emulate this partial input $\boldsymbol{X}_{\neg i}$ by introducing an attention mask to zero-out the attention to $\boldsymbol{x}_i$ in every layer (thus, the original position information holds). $q(y|\boldsymbol{X})$ represents the model's prediction probability for the token y_t given input $\boldsymbol{X}$.

Gradient norm (GN) GN (Simonyan et al., 2014; Li et al., 2016a) calculates the attribution score for each input token x_i by computing the L1 norm of its gradient of the target token y_t:

$$S_{\mathrm{GN}}(\boldsymbol{x}_i, y_t; \boldsymbol{X}) = \|g(\boldsymbol{x}_i, y_t; \boldsymbol{X})\|_{\mathrm{L1}} \quad (2)$$

$$g(\boldsymbol{x}_i, y_t; \boldsymbol{X}) = \nabla_{\boldsymbol{x}_i} q(y_t|\boldsymbol{X}), \quad (3)$$

where $g(\boldsymbol{x}_i, y_t; \boldsymbol{X}) \in \mathbb{R}^d$ denotes the gradient of the prediction probability for y_t with respect to $\boldsymbol{x}_i$, under the given input embedding sequence $\boldsymbol{X}$.

Input×gradient (I×G) I×G (Shrikumar et al., 2017; Denil et al., 2014) takes the dot product of a gradient with the respective token embedding $\boldsymbol{x}_i$:

$$S_{\mathrm{I\times G}}(\boldsymbol{x}_i, y_t; \boldsymbol{X}) = g(\boldsymbol{x}_i, y_t; \boldsymbol{X}) \cdot \boldsymbol{x}_i. \quad (4)$$

Integrated gradients (IG) IG (Sundararajan et al., 2017) is computed by accumulating gradients along a straight path from a baseline input $\boldsymbol{X}'$ to the actual input $\boldsymbol{X}$:

$$S_{\mathrm{IG}}(\boldsymbol{x}_i, y_t; \boldsymbol{X}) = (\boldsymbol{x}_i - \boldsymbol{x}'_i) \times \int_0^1 \frac{\partial q(y_t|\boldsymbol{X}' + \alpha(\boldsymbol{X} - \boldsymbol{X}'))}{\partial \boldsymbol{x}_i} \, d\alpha, \quad (5)$$

where $\boldsymbol{X}' := [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_n]$ denotes the sequence of baseline embeddings[2], and α denotes the interpolation coefficient. In practice, the integral is approximated using numerical integration with a finite number of steps.

Contrastive explanations For the IE, GN, and I×G methods, we adopt a contrastive explanation setting, which Yin and Neubig (2022) have shown to be quantitatively superior to the original non-contrastive setting. IA methods in this setting measure how much an input token x_i influences the model to increase the probability of target token y_t while decreasing that of foil token y_f. A foil token can be defined as an output with an alternative, incorrect generalization (§ 3). Contrastive versions of IE, GN, and I×G are defined as follows:

$$\begin{aligned} &S^*_{\mathrm{IE}}(x_i, y_t, y_f; \boldsymbol{X}) \\ &\quad = S_{\mathrm{IE}}(x_i, y_t; \boldsymbol{X}) - S_{\mathrm{IE}}(x_i, y_f; \boldsymbol{X}) && (6) \\ &S^*_{\mathrm{GN}}(\boldsymbol{x}_i, y_t, y_f; \boldsymbol{X}) = \|g^*(\boldsymbol{x}_i, y_t, y_f; \boldsymbol{X})\|_{\mathrm{L1}} && (7) \\ &S^*_{\mathrm{I\times G}}(\boldsymbol{x}_i, y_t, y_f; \boldsymbol{X}) = g^*(\boldsymbol{x}_i, y_t, y_f; \boldsymbol{X}) \cdot \boldsymbol{x}_i && (8) \\ &g^*(\boldsymbol{x}_i, y_t, y_f; \boldsymbol{X}) = \nabla_{\boldsymbol{x}_i}\big(q(y_t|\boldsymbol{X}) - q(y_f|\boldsymbol{X})\big) && (9) \end{aligned}$$

[2]We followed the common practice and employed a sequence of zero vectors as the baseline input. We used an interpretability library captum (Kokhlikyan et al., 2020) to calculate the IG score and keep all parameters as default.

2.2 Interpreting in-context learning (ICL)

We focus on the ICL setting (Brown et al., 2020), which has typically been adopted in modern LLM-based reasoning. An input prompt in ICL setting consists of few-shot examples E and a target question. E is composed of n examples $[e_1, \cdots, e_n]$, each of which contains an input-output pair $e_i = (X_i, f(X_i))$, given a function f associated with the task. Let X_{n+1} represent the target question q that the model must answer. The ICL setting is formed as follows:

$$\underbrace{\overbrace{\overbrace{X_1, f(X_1)}^{\textit{example } e_1}, \ldots, \overbrace{X_n, f(X_n)}^{\textit{example } e_n}}^{\textit{few-shot examples } E}, \overbrace{X_{n+1}}^{q},}_{\textit{prompt}} \underbrace{f(X_{n+1})}_{\textit{completion}}$$

Here, a model is expected to first induce the underlying function (rule) f from examples E and then generate the final output $f(X_{n+1})$.

Aha example Interpreting a model's in-context learning (ICL) involves identifying when, within the input, the model infers the correct rule f. To address this aspect, we propose a unique benchmark that features an explicit "aha moment" (e^*) within the input prompt. At this moment, the correct rule f can be identified by comparing the aha example with one of the other examples in the prompt. Thus, at least, e^* should be one of the two most important examples (see evaluation metrics in § 4.2). Note that, to mitigate the potential confusion, we exclude the case of e^* being the first example in the demonstration since, in this case, its next example e_2 can disambiguate the rule and virtually work as the aha example from the perspective of the incremental reasoning process.

Instance-level attribution Notably, we consider the use of IA methods to identify a particular example $e^* \in E$ in input, instead of a particular token. To compute an IA score for an example $S(e_i)$, we sum up the IA scores for its constituent tokens: $S(e_i) = \sum_{x_j \in (X_i, y_i) = e_i} S(x_j)$.[3] Our interest is which example obtains the highest IA score, i.e., $\arg\max_{e_i \in E} S(e_i)$.

3 Problem settings

We evaluate the performance of each IA method in identifying the crucial in-context example e^*

[3]An exception applies in the IE method; the attribution score for an example e_i is simply computed by erasing the corresponding X_i and $f(X_i)$ from the input sequence.

Task	Prompt example/template	Answer	Potential rules
LINEAR-OR-DISTINCT	a a b a ↦ b g g j g ↦ j k i k k ↦ k / i o o o p ↦	o / p	A. Generate the n-th token (3rd token in this example) B. Generate the distinctive token
ADD-OR-MULTIPLY	aa ↦ aaaa hh ↦ hhhh vvv ↦ vvvvv / vvvvvv i ↦	iii / ii	A. Add m tokens ($m = 2$ in this example) B. Multiply the numder of tokens by n ($n = 2$ in this example)
VERB-OBJECT	*like* [CITY] ↦ True *love* [ANIMAL] ↦ False *like* [ANIMAL] ↦ True / False *love* [CITY] ↦	False / True	A. If "like" exists, then True B. If [CITY] exists, then True
TENSE-ARTICLE	*The* [NOUN] [VERB]-ing ↦ True *A* [NOUN] [VERB]-past ↦ False *A* [NOUN] [VERB]-ing ↦ True / False *The* [NOUN] [VERB]-past ↦	False / True	A. If the verb is in ing form, then True B. If the first token is "the", then True
POS-TITLE	The [NOUN] Was [ADJ] ↦ True The [noun] was [noun] ↦ False The [noun] was [adj] ↦ True / False The [NOUN] Was [NOUN] ↦	False / True	A. If adjective exsist, then True B. If the sentence is in title case, then True
ASSOCIATIVE-RECALL	a ↦ 6 g ↦ 3 w ↦ 5 g ↦	3	Key–value pairs are in the prompt. The task is to output a value associated with a given key.

Table 1: Formats of our inductive reasoning tasks. As a baseline setting, we also set ASSOCIATIVE-RECALL setting to just memorize key-value mappings. The remaining tasks span from somewhat superficial features to linguistic ones. The disambiguating example (the third one in these examples) determines the correct rule and answer (blue or orange) for the final question from two plausible generalizations shown in the "Potential rules" column.

necessary for defining the task. In real-world tasks, it is generally unclear which in-context example is the most influential in solving the task, and the task may be solved even without relying on any of the examples (e.g., solved by leveraging prior knowledge). Therefore, these are not suitable as a benchmark to evaluate the interpretability method, and we design a synthetic and controlled tasks.

Our setting is the extension of Mueller et al. (2024); we employ a set of ambiguous inductive learning scenarios inspired by the cognitively-motivated LM analyses (McCoy et al. 2020,Warstadt et al. 2020a; *inter alia*). In these scenarios, a task f is mostly ambiguous in demonstrations E in the sense that several compatible rules exist to explain the transformations $X \mapsto f(X)$. We extend this setting by adding only one disambiguating example e^* ("*aha example*"), which determines the correct rule f^* to be unique, and test whether each IA method can identify this special example as long as models correctly employ this clue e^* to resolve the problem. For instance, most examples shown in Figure 1 are ambiguous (with gray color) w.r.t. the two possible rules of (i) adding the same token twice or (ii) multiplying the number of tokens by two. This ambiguity is resolved by comparing the aha example e^* (blue example in Figure 1) with any one of the other ambiguous examples. As shown in Table 1, we designed the following tasks as a case study:

LINEAR-OR-DISTINCT (LD) The few-shot examples are ambiguous as to Rule A: selecting a character in a particular linear position in an input X_i; or Rule B: selecting a character that differs from the others in an input X_i.

ADD-OR-MULTIPLY (AM) The ambiguity of this task is Rule A: add a certain number of tokens to input X_i; or Rule B: multiply the numbers of tokens in the input X_i.

VERB-OBEJECT (VO) This task requires distinguishing whether the type of verb (Rule A) or the category of the object noun (Rule B) matters. We employed two verbs ("like" and "love") and two categories of the object (`city` or `animal`).

TENSE-ARTICLE (TA) The potential rules are Rule A: whether the main verb in the input X_i is in ing-form or not; or Rule B: whether the first token of input X_i is "The" or not.

POS-TITLE (PT) This task involves two rules: Rule A: whether there is an adjective in X_i; or Rule B: whether X_i is presented in the title case.

In addition to them, we adopted a simple task of associative recall (AR), which is typically employed in studying ICL, where the model is supposed to simply memorize the key: value mapping rules demonstrated in the prompt and apply them to the target question. Linguists may be more interested in the task of, for example, syntactic transformation to an interrogative sentence (McCoy et al., 2020) based on the original poverty of the stimulus argument in the language domain (Chomsky, 1980). However, such a realistic task interferes with the models' meta-linguistic knowledge; thus, we adopted artificial (and somewhat simpler) ones.

Foil token A contrastive explanation needs a foil token corresponding to an explicit negative label (§ 2.1). We use the token/answer corresponding to an alternative rule (conflicting the disambiguating example) as the foil token.

4 Experimental setup

4.1 Overview

Few-shot settings We conducted experiments with different numbers of few-shot examples; specifically, we examined 10-shot, 50-shot, and 100-shot settings to test the robustness of IA methods toward somewhat longer demonstrations.

Data For each synthetic task, we create 360 different questions with different sets of few-shot examples and a target question. In the LD, AM, VO, TA, and PT tasks, the correct rule is selected out of the two candidates (rules A or B shown in Table 1) in a 1:1 ratio. The position of the most influential (i.e., disambiguating) example e^* is assigned according to a uniform distribution over all positions except the first. We test IA methods using only the questions that models answered correctly.

Models We evaluate five LLMs: Llama-2-7B, Llama-2-13B (Touvron et al., 2023), Gemma-2-2B, Gemma-2-9B, and Gemma-2-27B (Riviere et al., 2024). As a prerequisite of our experiments, the models should be able to learn the task, i.e., be sufficiently sensitive to the disambiguating example e^* and use this to determine the correct rule. To ensure this ability, we fine-tune these models on each task (see Appendix A), but the conclusions overall did not alter before and after fine-tuning (see Appendix B).

4.2 Metrics

We report two accuracy measures: (i) e^* is in the top two examples with the highest IA score (top-2 accuracy), and (ii) e^* gets the highest IA score among the input examples (top-1 accuracy). Top-2 accuracy is motivated by the fact that models should at least consider the e^* plus any other example to identify the correct rule (as described in § 2.2). Top-1 accuracy is motivated from a leave-one-out perspective; excluding the e^* significantly hurts the task answerability, while excluding the other examples does not hurt the task ambiguity/complexity.

4.3 Baselines

Along with the IA methods introduced in § 2.1, we evaluate four baseline methods.

Edit distance This method identifies e^* simply using edit distance between the target example $X_{n+1} \oplus y_{n+1}$ and each example $X_i \oplus y_i$, where $\oplus$ is a string concatenation. Example with the minimum edit distance, thus the most similar example to the target question, is selected as an explanation. The weak performance of this problem probes that our experimental setting is so challenging that just relying on surface features does not resolve it.

Attention weights This method leverages attention weights, computed as the sum of attention weights across all tokens in input X. While attention weights are generally considered unreliable for model interpretation, we include this baseline to compare whether IA methods achieve superior performance.

Self-answer We also examine directly asking the models to generate their rationale. Specifically, we have models generate the most informative example in a prompt (Appendix D) in deriving their answer as a post-hoc explanation. This might be, more or less, relevant to the verbalization of *aha moment* recently observed in DeepSeek models (Guo et al., 2025).

Chance rate We also report the chance rate of attribution accuracy when randomly selecting one example from a prompt.

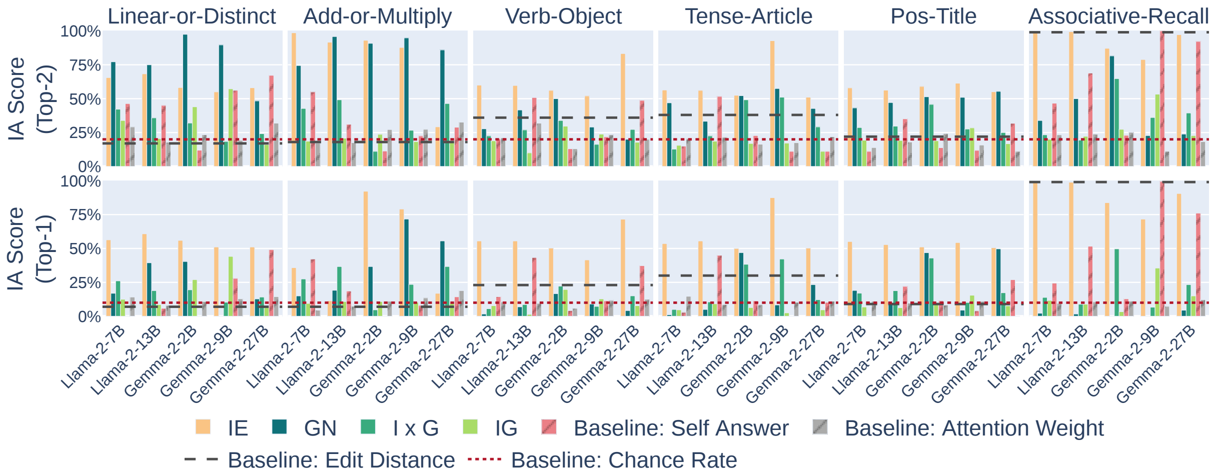

Figure 2: Attribution accuracies for each task/model in the 10-shot setting (thus, the chance rate is 20% and 10% for the top-two and top-one metric, respectively; red dotted line). The edit distance and attention baselines are indicated by a black dotted line and gray bar, respectively.

5 Experiments

Figure 2 shows the results in 10-shot settings, with both top-2 and top-1 metrics. Additional analyses are presented in Appendices C, E, and F.

5.1 Main results

IE works the best First of all, the input erasure method generally performed the best in both top-1 and top-2 accuracies. This is somewhat obvious because our task is designed to be unsolvable by removing the aha example and thus rather serves as a quick check for our experimental design. Having said that, the input erasure method has some disadvantages in regard to the computational costs of repeated decoding by removing examples one by one as well as the unclarity of by which unit an erasure should be applied, especially in a real, somewhat noisy input. Additionally, the accuracy was not 100% in almost all the cases; we further discuss the potential flaw of this approach in § 6.

Potential of gradient-based approaches As for baselines, while the self-answer approach worked well in specific settings (associative recall with larger models), most baselines, including attention weights, generally failed to achieve high accuracy. Edit distance was a somewhat strong baseline, but it has obvious limitations of lacking semantic similarities and was frequently outperformed by GN. Compared to such baselines, the gradient-based method worked relatively well, highlighting the potential of this direction.

Improved versions of gradient-based methods do not outperform GN Among the gradient-based methods, simple gradient norm tends to work the best in most tasks, especially in top-2 accuracy. In other words, whereas I×G and IG are proposed as the improved version of simple gradient norm method, there were no substantial advantages of these methods in our settings. In particular, IG consistently yielded the lowest attribution accuracy across all six tasks among the gradient-based methods, suggesting its limitations in ICL scenarios. The plausible reason behind this inferiority is discussed in § 6.

General failure Nevertheless, some simple tasks, such as VERB-OBJECT, TENSE-ARTICLE, and POS-TITLE, were ever hard to interpret with any approach. This opens a new field for developing a better interpretation method for ICL.

5.2 Scaling properties

In the age of LLMs, the setting has progressively been scaled up toward model parameter size and context length. We analyze how such a scaling affects the LLMs' interpretability.

Interpretability vs. model size We first investigate the relationship between attribution accuracy and model size — is it more difficult to interpret larger models? We observe somewhat intriguing patterns for this question (Figure 2); gradient-based methods tend to work worse in larger models, and in contrast, the self-answer baseline works better in larger models (especially in LINEAR-OR-DISTINCT and ASSOCIATIVE-RECALL). That is,

Task	Accuracy (%)	IE Attr. Acc. (%)	
		Top-2	Top-1
LD (Rule A)	98.0	58.2	56.0
LD (Rule B)	0.5		
AM (Rule A)	34.5	93.1	92.2
AM (Rule B)	65.5		
VO (Rule A)	100.0	56.1	50.3
VO (Rule B)	0.0		
SP (Rule A)	98.0	52.5	50.0
SP (Rule B)	2.0		
ST (Rule A)	68.5	59.0	51.1
ST (Rule B)	31.5		

Table 2: Task accuracy (not attribution accuracy) of Gemma-2-2B (excluding AR) when the disambiguating example is not included, separated by the correct rule. The accuracy drastically differs when the correct rule is different; thus, the models adopt a particular default rule with their inductive biases against fully ambiguous demonstrations, even in our controlled settings.

the (empirically) accurate approach to interpreting the LLMs may differ in their model scale, and the success in interpreting smaller models does not always entail the success in interpreting larger models.

Interpretability vs. number of examples Next, given the trend of long-context LLMs, we examine the relationship between attribution accuracy and the number of few-shot examples. Figure 3 shows the attribution accuracy for Gemma-2-2B across all six tasks in different number of in-context examples. This demonstrates that gradient-based methods maintain accuracy or rather improve against the longer context, in contrast to the decreasing chance rate. This suggests the robustness of IA methods in long-context scenarios, highlighting their potential for interpreting inputs with extensive contextual information. Notably, the quality of self-answer consistently degraded as the number of in-context examples increased, while the positive scaling effect was observed in the previous analysis of model size. That is, the gradient-based methods and self-answer approach exhibit an insightful trade-off between different scaling properties.

6 Discussion

This section discusses the potential reasons for the unexpected results presented in § 5, highlighting the challenging issues in interpreting ICL.

Why did IE fail to achieve 100% attribution accuracy? Our tasks can not be answered without disambiguating *aha* examples. Thus, it is somewhat unintuitive to see the non-100% attribution accuracy of the IE method (again, the LLMs understood the task as they achieved 100% accuracy in the tasks) — what happens here? To obtain a hint to clarify IE's potential limitations, we analyze model behaviors when the disambiguating example is excluded. Interestingly, LLMs adopted a specific generalization (rule) in each task when there was no disambiguating example (Table 2); in other words, they sometimes exhibited strong inductive biases in our tasks. That is, when the correct rule is equal to their preferred rule by their inductive bias, they can answer the task correctly even without disambiguating examples, and the IE method does not compute a proper attribution score. It is now common to see LLMs have particular inductive biases (not a *tabula rasa*) (Warstadt et al., 2020b; Kharitonov and Chaabouni, 2021). Catching such generalization bias with IA methods represents an inherent challenge, highlighting their potential limitations in interpreting ICL.

Why was I×G worse than gradient norm? One advantage of the I×G method, compared to the simple gradient norm, is the consideration of the norm of the input embedding (Shrikumar et al., 2017; Denil et al., 2014). Since a large vector tends to have a large dot product with another vector, the norm of the input token embedding (vector) is expected to affect the IA score of I×G. Then, no improvement of I×G over the simple gradient norm suggests that, at least in our settings, the norm of the embeddings was not informative to estimate the input attribution. The norm of the embedding largely has decontextualized information about the word, such as frequency, and it may make sense that such information is not helpful to interpreting our controlled, artificial ICL tasks consisting of alphabet characters, numbers, or random words.

Why was IG worse than gradient norm? IG is a path-based approach; the gradient is accumulated from a baseline vector (typically zero vector) to the targeted input representation (in our case, the sequence of input embeddings representing few-shot examples). This approach is somewhat intuitive when considering an attribution for a particular word or sentence; for example, suppose one computes an attribution to the word "excellent" in an input for a particular task-specific model, IG may trace the path from zero to the "excellent" vector, which will be in a kind of the *goodness* direction, involving the points corresponding to,

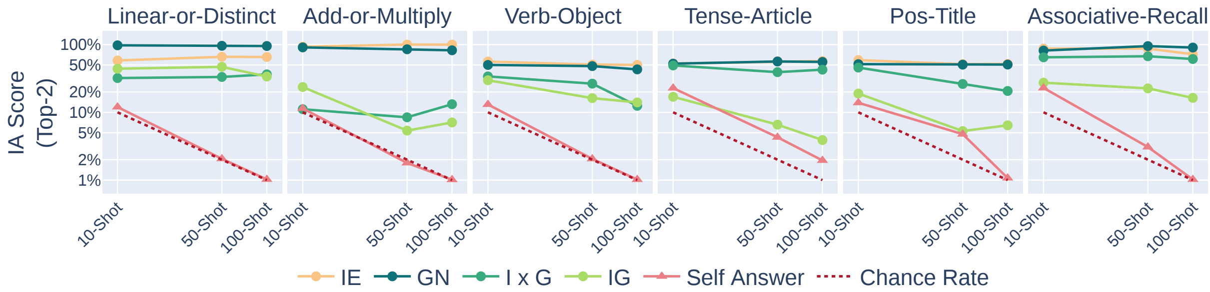

Figure 3: Attribution accuracy for interpreting Gemma-2-2B models across all six tasks. Gradient-based methods are relatively robust to the number of few-shot examples, while there is a consistent, large drop in attribution accuracy in SA. Note that both x-axis and y-axis are in log scale.

e.g., "okay" "decent," "good," "excellent" (Sanyal and Ren, 2021). Then, one critical question is — what does this path mean in the prompt/task space? Different prompt representations will no longer correspond to the same task; thus, the attribution of a particular token in the middle of such a path in the prompt space may no longer be an attribution under a targeted task. This can be one concern toward the ineffectiveness of IG in our settings.

7 Related Work

IA methods Several lines of research are conducted to interpret neural language models. NLP researchers have adapted IA methods, which were originally applied to vision models (Simonyan et al., 2014; Springenberg et al., 2015; Zintgraf et al., 2017), to perform a post-hoc interpretation of input-output associations exploited by language models (Karpathy et al., 2015; Li et al., 2016a; Arras et al., 2016; Lei et al., 2016; Alvarez-Melis and Jaakkola, 2017), and its improved versions have also been developed (Denil et al., 2014; Sundararajan et al., 2017; Murdoch et al., 2018; Sinha et al., 2021; Ding and Koehn, 2021; Bastings et al., 2022; Yin and Neubig, 2022; Ferrando et al., 2023). In line with these studies, we provide a new perspective to evaluate these IA methods in ICL. Note that, as an orthogonal attempt, some research estimates the saliency scores to directly prompt models to generate such explanations (Rajani et al., 2019; Liu et al., 2019; Wu and Mooney, 2019; Narang et al., 2020; Marasovic et al., 2022). This method is indeed examined as one baseline in our study.

Instance-based explanation Instance-based explanation seeks for the explanation in training data rather than the immediate input during inference as IA methods (Wachter et al., 2017; Charpiat et al., 2019; Hanawa et al., 2021). These two paradigms of instance-based and IA-based explanations have been studied somewhat separately since the information source to seek the explanation is clearly different. On the other hand, in ICL, the training examples are now in the input during inference that can be analyzed by IA methods. In this sense, our investigation can seen as a new exploration of instance-based explanation with the help of IA methods.

Mechanistic Interpretability With the rise of large language models, such as GPT-3 (Brown et al., 2020), the mechanistic interpretability community has shifted its focus from vision models to language models. Within which, the promising results using sparse autoencoders (SAEs) (Bricken et al., 2023; Templeton et al., 2024) have inspired a flurry of follow-up work (Gao et al., 2024; Lieberum et al., 2024; Rajamanoharan et al., 2024a,b; Karvonen et al., 2024; Braun et al., 2024; Kissane et al., 2024; Makelov, 2024). Such a scope of SAE, interpreting the model internals, is orthogonal to our direction of estimating the importance of input examples.

8 Conclusions

We have pointed out and tackled the problem of interpreting the inductive reasoning process in ICL as a missing but reasonable milestone to be explored in LLM interpretability research. Our revisit to the IA methods in interpreting this ICL process has clarified their limitations from a new angle as well as provided fruitful insights and discussions on their practical usage in modern NLP. These findings have highlighted some issues in the community; in particular, even the fundamental task of mapping input and output has not been accomplished, and there is room to sophisticate previously developed interpretability tools to be suitable for LLMs.

Limitations

Our study has several limitations in scope. First, we focused primarily on popular gradient-based IA methods, leaving other approaches such as perturbation-based methods like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) for future work.

Regarding model selection, we concentrated on widely-used open-weight LLMs. Since applying IA methods requires gradient computation through backward propagation, computational constraints limited our ability to evaluate all available models, particularly large ones such as Llama-2-70B (Touvron et al., 2023).

Our experimental design used synthetic tasks to better identify influential examples in the few-shot setting. While this approach allowed for controlled experimentation, both the number and format of tasks were limited. Future work could explore more realistic tasks with greater variations.

We focused exclusively on pre-training models, excluding post-training models from our analysis. This choice was motivated by our interest in basic few-shot learning, which is more commonly used with pre-trained models. Although post-training models might demonstrate higher accuracy on our tasks and the self-answer setting due to their potentially superior capabilities (Riviere et al., 2024), our primary focus was on evaluating IA methods rather than model performance.

The optimization of the self-answer setting was not explored in depth, as our main interest lay in examining whether larger models showed improved performance in this setting rather than enhancing the setting itself.

Finally, while our ambiguous tasks were designed with two potential functions in mind, we acknowledge that models might interpret these tasks differently than intended. However, since our focus was on whether models could recognize these as ambiguous tasks with two possible answers and use specific examples to determine the appropriate response, we believe this limitation does not significantly impact our findings.

Ethical Statements

This work advances our understanding of input attribution (IA) methods in the context of large language models' (LLMs) in-context learning (ICL). Our findings contribute to the broader goal of developing more interpretable and safer AI systems by providing practical insights into the strengths and weaknesses of IA methods as tools for interpreting LLMs.

This study exclusively uses synthetic data generated through computational methods. No real user data, human annotations, or personally identifiable information was collected or used in our experiments. Our synthetic dataset generation process does not involve any human subjects, crowd workers, or demographic information.

References

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark. Association for Computational Linguistics.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, and 30 others. 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany. Association for Computational Linguistics.

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy. Association for Computational Linguistics.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic Interpretability for AI Safety - A Review. *Transactions on Machine Learning Research (TMLR)*.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying Functionally Important Features with End-to-End Sparse Dictionary Learning. *Preprint*, arXiv:2405.12241.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. 2019. Input Similarity from the Neural Network Perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5343–5352.

Noam Chomsky. 1980. *Rules and Representations*. Columbia University Press.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2014. Extraction of Salient Sentences from Labelled Documents. *Preprint*, arXiv:1412.6815.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *Preprint*, arXiv:2406.04093.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. 2021. Evaluation of Similarity-based Explanations. In *International Conference on Learning Representations (ICLR)*.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and Understanding Recurrent Networks. *Preprint*, arXiv:1506.02078.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Riggs Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. 2024. Measuring Progress in Dictionary Learning for Language Model Interpretability with Board Game Models. In *ICML Workshop on Mechanistic Interpretability*.

Eugene Kharitonov and Rahma Chaabouni. 2021. What they do when in doubt: a study of inductive biases in seq2seq learners. In *International Conference on Learning Representations (ICLR)*.

Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. 2024. Interpreting Attention Layer Outputs with Sparse Autoencoders. In *ICML Workshop on Mechanistic Interpretability*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. *Preprint*, arXiv:2009.07896.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016b. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016c. Understanding Neural Networks through Representation Erasure. *Preprint*, arXiv:1612.08220.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca D. Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma Scope: Open Sparse

Autoencoders Everywhere All At Once on Gemma 2. *Preprint*, arXiv:2408.05147.

Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581, Florence, Italy. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4765–4774.

Aleksandar Makelov. 2024. Sparse Autoencoders Match Supervised Features for Model Steering on the IOI Task. In *ICML Workshop on Mechanistic Interpretability*.

Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew E. Peters. 2022. Few-Shot Self-Rationalization with Natural Language Prompts. In *Findings of the Association for Computational Linguistics (NAACL)*, pages 410–424.

R Thomas McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Annual Meeting of the Cognitive Science Society (CogSci)*, pages 2096–2101.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. 2024. In-context learning generalizes, but not always robustly: The case of syntax. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4761–4779.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *International Conference on Learning Representations (ICLR)*.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. WT5?! Training Text-to-Text Models to Explain their Predictions. *Preprint*, arXiv:2004.14546.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom In: An Introduction to Circuits. *Distill*.

Amy Perfors, Joshua B. Tenenbaum, and Terry Regier. 2011. The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338.

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350, Melbourne, Australia. Association for Computational Linguistics.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. 2024a. Improving Sparse Decomposition of Language Model Activations with Gated Sparse Autoencoders. In *ICML Workshop on Mechanistic Interpretability*.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramar, and Neel Nanda. 2024b. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders. *Preprint*, arXiv:2407.14435.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 178 others. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *Preprint*, arXiv:2408.00118.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10285–10299.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning (ICML)*, pages 3145–3153.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *International Conference on Learning Representations Workshop Track (ICLR)*.

Sanchit Sinha, Hanjie Chen, Arshdeep Sekhon, Yangfeng Ji, and Yanjun Qi. 2021. Perturbing inputs for fragile interpretations in deep natural language processing. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 420–434, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. In *International Conference on Learning Representations (ICLR)*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning (ICML)*, pages 3319–3328.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *Preprint*, arXiv:2307.09288.

Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *Preprint*, arXiv:1506.05869.

Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Preprint*, arXiv:1711.00399.

Alex Warstadt and Samuel R. Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, pages 1737–1743.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020a. Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Colin Wilson. 2006. Learning phonology with substantive bias: an experimental and computational study of velar palatalization. *Cogn. Sci.*, 30(5):945–982.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Jialin Wu and Raymond Mooney. 2019. Faithful multimodal explanation for visual question answering. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 103–112, Florence, Italy. Association for Computational Linguistics.

Aditya Yedetore, Tal Linzen, Robert Frank, and R Thomas McCoy. 2023. How poor is the stimulus? Evaluating hierarchical generalization in neural networks trained on child-directed speech. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9370–9393.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In *International Conference on Learning Representations (ICLR)*.

A Finetuning details

The fine-tuning dataset consisted of 400 tasks for each of the 10-shot, 50-shot, and 100-shot settings (1,200 tasks total). For each task, we created a training set for fine-tuning using tokens that did not overlap with our test set (the dataset used in our main experiments). We fine-tuned models separately on each task, resulting in six fine-tuned models per LLM. The exception was the Gemma-2-27B model, which we did not fine-tune on the ASSOCIATIVE-RECALL task since the original model already performed well enough on this simple task.

A.1 Finetuing parameters

We use a consistent LoRA configuration with rank $r = 32$ and scaling factor $\alpha = 64$, applying a dropout rate of 0.05 across all linear modules. The LoRA adaptation includes bias terms in the training. For optimization, we perform a learning rate sweep using a cosine scheduler with a 5% warm-up period relative to the total training steps. The optimal learning rate typically falls in the order of 1×10^{-5} when the loss reaches its minimum. In our experiments, the Llama-2 models achieved nearly zero loss, which is expected in such a synthetic setting. The Gemma-2 models, however, converge to final loss values of approximately 0.2.

A.2 Zero-shot task accuracy after finetuning

We evaluate task accuracy using exact match, with results presented in Tables 4, 5, 6, 7, and 8. In the zero-shot setting, some tasks show accuracies significantly below chance rate (indicated in parentheses), as models occasionally generate unexpected responses. Notably, all models achieve zero-shot accuracies at or below the chance rate across all tasks, suggesting that models cannot solve our tasks relying only on the aha example.

B Base model results

Figure 4 presents the IA scores for the base models. While the overall IA scores for VO, TA and PT tasks are relatively low, the performance trends across different tasks and models exhibit similar patterns to those observed in the fine-tuned models (Figure 2). Therefore, our results can be generalized regardless of fine-tuning.

C Aggregating attritbuion with *max*

Figure 5 presents the IA scores using maximum aggregation to convert token-level attribution to example-level attribution: $S(e_i) = \max_{x_j \in (X_i, y_i) = e_i} S(x_j)$. The overall trend for all the IA methods is consistent with the sum aggregation (Figure 2); thus, our results can be generalized regardless of this design.

D Prompts

We present sample of the exact prompt we used for our task, including the ones we used for testing attribution accurices and modified prompts for self-answer. Note that in all the experiments, we only used the model outputs with a correct answer. That is why we appended the correct answer in advance to the self-answer prompt to obtain the post-hoc explanation.

Normal Prompt

```
Input: they, Output: 6
Input: not, Output: 3
Input: I, Output: 5
Input: tell, Output: 7
Input: them, Output: 6
Input: were, Output: 6
Input: at, Output: 0
Input: yes, Output: 1
Input: right, Output: 9
Input: say, Output: 3
Input: they, Output:
```

Self-answer Prompt

```
<0>Input: they, Output: 6</0>
<1>Input: not, Output: 3</1>
<2>Input: I, Output: 5</2>
<3>Input: tell, Output: 7</3>
<4>Input: them, Output: 6</4>
<5>Input: were, Output: 6</5>
<6>Input: at, Output: 0</6>
<7>Input: yes, Output: 1</7>
<8>Input: right, Output: 9</8>
<9>Input: say, Output: 3</9>
<target>Input: they, Output: </target>

Among the 10 examples labeled <0> to <9>, select the single most helpful example for determining the answer to the <target> question. The correct answer to the target question is "6". To conclude this answer, we need to find one example that provides the necessary information. Therefore, the most helpful example is <
```

E Chain-of-though format

To contextualize our experimental settings with more practical scenarios, we further evaluate attribution accuracies on top of chain-of-thought (CoT) prompting (Wei et al., 2022). We use Gemma-2-27B-IT (a post-training version of Gemma-2-27B) instead of the base model to perform a better CoT-style generation and employ the AM task, where the model achieved high accuracy with CoT, as a case study. We only target the last time step to generate the exact answer. We compute the by-example attribution scores the same as our main experiments, but now the attribution scores can be spread over the reasoning chain part as well as in-context examples. Our target is which example is informative to answer the question; thus, the attribution to the chain part is tentatively disregarded. As statistics, we just report how many proportions of attribution scores ([0-100%]) reached the reasoning chain part (denoted as "Chain prop.").

The results are presented in Table 3. All tested IA methods performed worse in this CoT setting than in the CoT-free settings in the main experiments. Nevertheless, the superiority of GN to other approaches still holds. Note that the Chain prop. substantially differs across IA methods; for example, IE assigns over 80% of the attribution score to the chain. These divergent results also suggest that the conventional IA methods can not easily be applied to modern ICL and CoT settings.

The exact prompt and the reasoning chain generated by the model are provided below[4]:

CoT Prompt

<start_of_turn> user

Method	IA score Top-1	IA score Top-2	Aggregation	Chain prop. (%)
IE	11.3	17.4	Sum	82.1
GN	12.4	37.6	Sum	35.4
I × G	14.9	29.8	Sum	23.0
IG	8.9	22.7	Sum	42.9
IE	11.3	17.4	Max	82.1
GN	14.5	33.3	Max	19.2
I × G	9.9	31.2	Max	23.1
IG	9.9	22.3	Max	7.5

Table 3: IA score for the CoT-prompted AM task. The percentage of attribution scores allocated to the reasoning chain is denoted as Chain prop.

Input: saw, 2, Output: saw, 4
Input: start, 2, Output: start, 4
Input: the, 2, Output: the, 4
Input: too, 2, Output: too, 4
Input: round, 2, Output: round, 4
Input: which, 1, Output: which, 3
Input: work, 2, Output: work, 4
Input: get, 2, Output: get, 4
Input: that, 2, Output: that, 4
Input: white, 2, Output: white, 4
Input: I, 3, Output: <ANSWER>

Solve this problem step by step, generate the content of <ANSWER> after "So the answer is": <end_of_turn>

<start_of_turn> model

F Distribution of example with the highest attribution score

Figures 6, 7, 8, 9 and 10 present the distribution of example positions with the highest attribution scores. All IA methods, except for I × G, possess positional bias for certain tasks, specifically favoring examples either at the beginning or end, aligning with the position bias known to LLMs (Liu et al., 2024).

G Informaiton for responsibility checklist

We utilized software libraries, including the Huggingface toolkit (Wolf et al., 2020) (Apache License Version 2.0), Captum package (Kokhlikyan et al., 2020) (BSD 3-Clause License) for IG computation, and Pytorch (Ansel et al., 2024) (BSD 3-Clause License). These tools were used according to their licenses and intended usage. We used writing assistance tools (including Grammarly) for language error correction only. The computational

[4]The prompt template is applied since this is a post-training model

budget of this work is approximately 600 GPU hours (with A100/H100/H200 machines).

	Zero-shot (%)	Few-shot (%)
ASSOCIATIVE-RECALL	10.0	100.0 (10.0)
LINEAR-OR-DISTINCT	44.8	99.0 (50.0)
ADD-OR-MULTIPLY	11.8	100.0 (50.0)
VERB-OBJECT	0.0	100.0 (50.0)
TENSE-ARTICLE	0.0	100.0 (50.0)
POS-TITLE	0.0	98.0 (50.0)

Table 4: The zero-shot and few-shot accuracy of the fine-tuned Gemma-2-2B model across all evaluation tasks. The chance rate is indicated in parentheses.

	Zero-shot (%)	Few-shot (%)
ASSOCIATIVE-RECALL	12.0	100.0 (10.0)
LINEAR-OR-DISTINCT	50.0	85.5 (50.0)
ADD-OR-MULTIPLY	12.0	100.0 (50.0)
VERB-OBJECT	0.0	94.8 (50.0)
TENSE-ARTICLE	0.0	100.0 (50.0)
POS-TITLE	50.0	98.8 (50.0)

Table 5: The zero-shot and few-shot accuracy of the fine-tuned Gemma-2-9B model across all evaluation tasks. The chance rate is indicated in parentheses.

	Zero-shot (%)	Few-shot (%)
ASSOCIATIVE-RECALL	15.0	100.0 (10.0)
LINEAR-OR-DISTINCT	47.3	99.8 (50.0)
ADD-OR-MULTIPLY	48.5	97.3 (50.0)
VERB-OBJECT	0.0	99.5 (50.0)
TENSE-ARTICLE	50.0	100.0 (50.0)
POS-TITLE	45.5	97.8 (50.0)

Table 6: The zero-shot and few-shot accuracy of the fine-tuned Gemma-2-27B model across all evaluation tasks. The chance rate is indicated in parentheses.

	Zero-shot (%)	Few-shot (%)
ASSOCIATIVE-RECALL	10.0	100.0 (10.0)
LINEAR-OR-DISTINCT	44.8	100.0 (50.0)
ADD-OR-MULTIPLY	11.8	99.0 (50.0)
VERB-OBJECT	0.0	100.0 (50.0)
TENSE-ARTICLE	0.0	100.0 (50.0)
POS-TITLE	0.0	98.0 (50.0)

Table 7: The zero-shot and few-shot accuracy of the fine-tuned Llama-2-7B model across all evaluation tasks. The chance rate is indicated in parentheses.

	Zero-shot (%)	Few-shot (%)
ASSOCIATIVE-RECALL	10.0	100.0 (10.0)
LINEAR-OR-DISTINCT	50.0	99.8 (50.0)
ADD-OR-MULTIPLY	41.0	100.0 (50.0)
VERB-OBJECT	0.0	100.0 (50.0)
TENSE-ARTICLE	0.0	100.0 (50.0)
POS-TITLE	41.8	97.0 (50.0)

Table 8: The zero-shot and few-shot accuracy of the fine-tuned Llama-2-13B model across all evaluation tasks. The chance rate is indicated in parentheses.

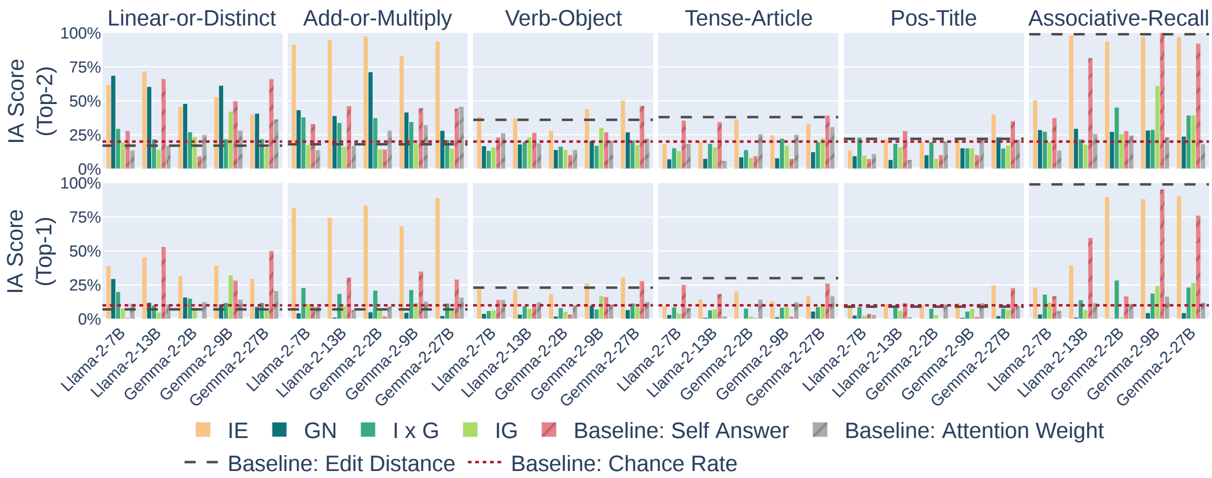

Figure 4: Attribution accuracies for each task for base models. Similar patterns to those observed in the fine-tuned models (Figure 2) can be obsered.

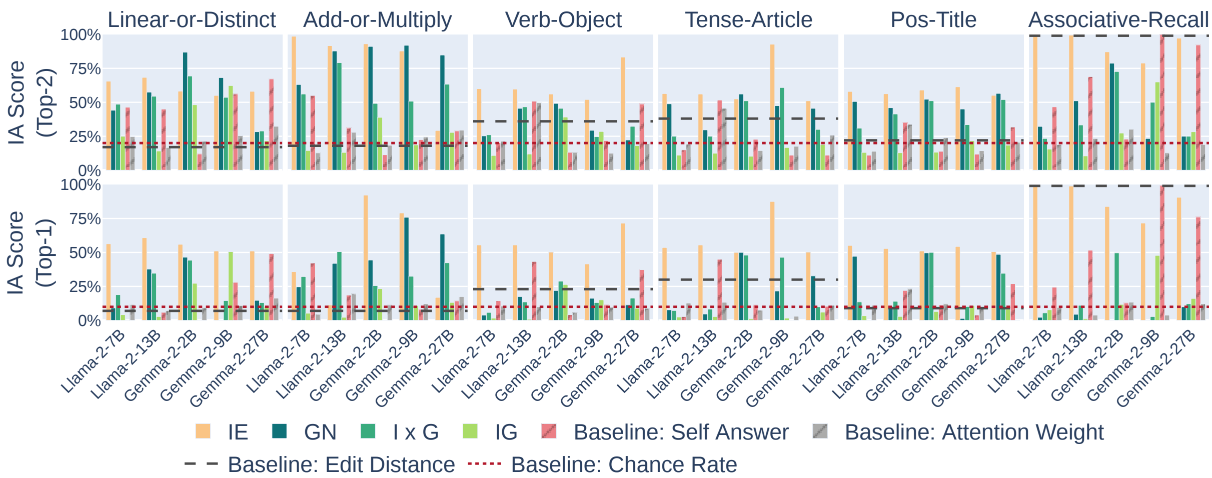

Figure 5: Attribution accuracies for each task use max aggregation. The overall trend for all IA methods is consistent with sum aggregation (Figure 2)

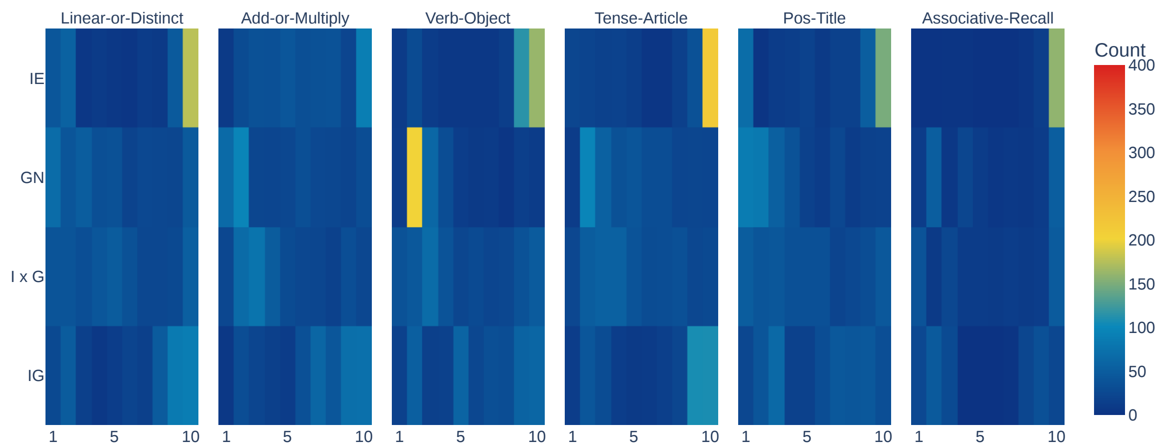

Figure 6: Distribution of the positional of the example with the highest attribution scores across IA methods (Llama-2-7B model).

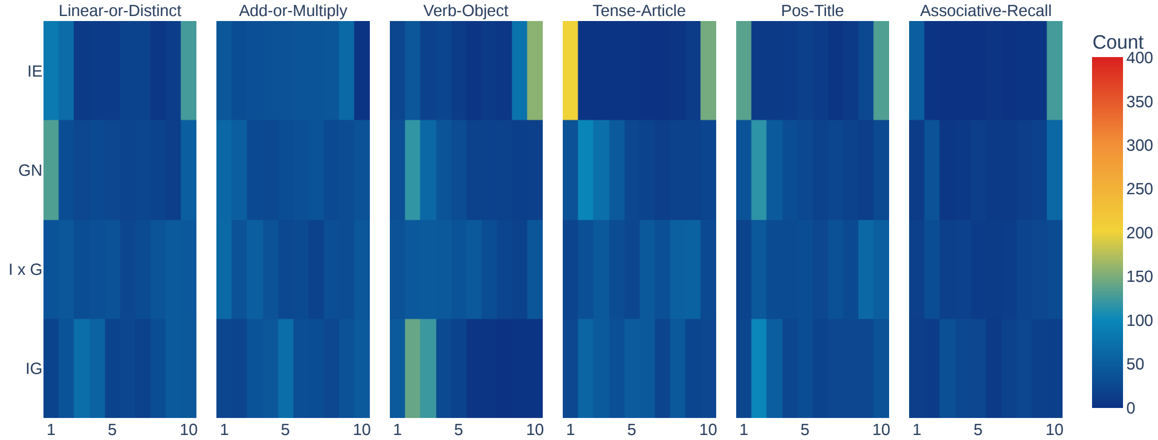

Figure 7: Distribution of the position of the example with the highest attribution scores across IA methods (Llama-2-13B model).

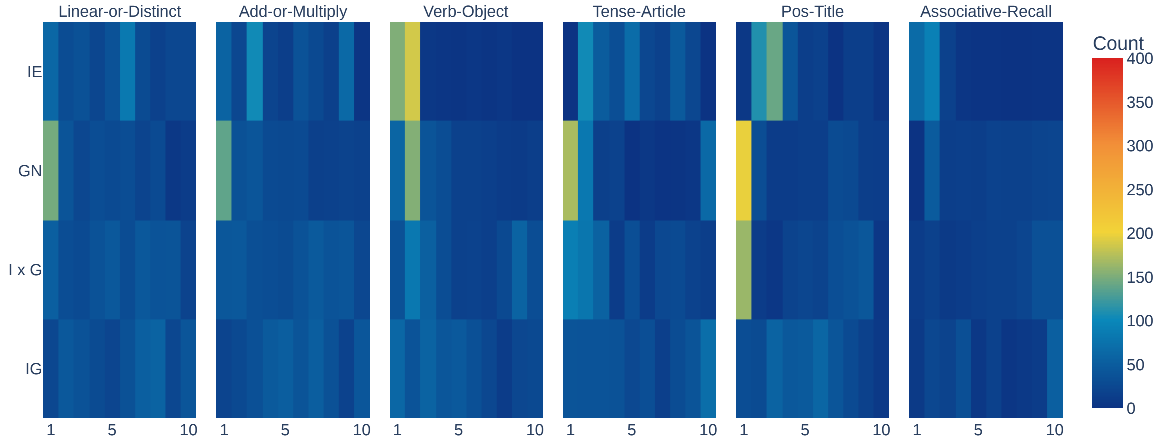

Figure 8: Distribution of the position of the example with the highest attribution scores across IA methods (Gemma-2-2B model).

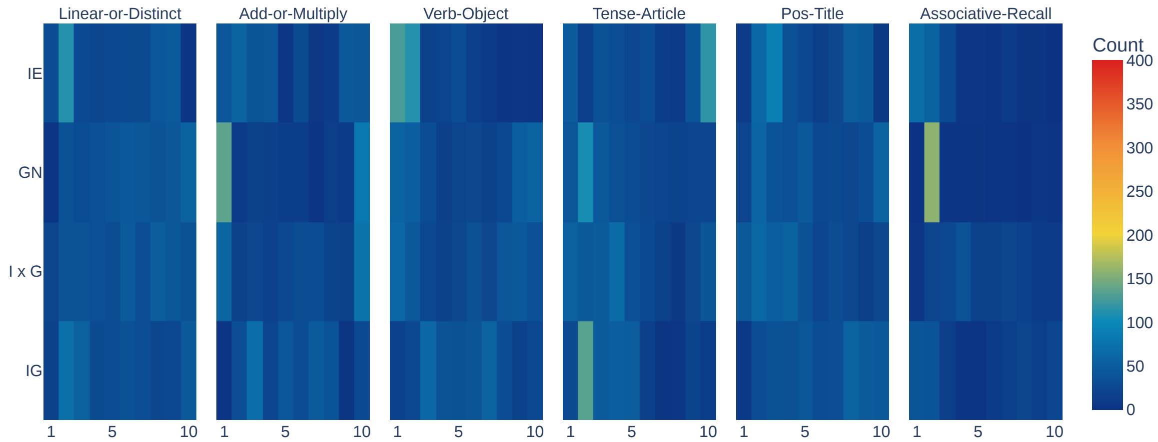

Figure 9: Distribution of the position of the example with the highest attribution scores across IA methods (Gemma-2-9B model).

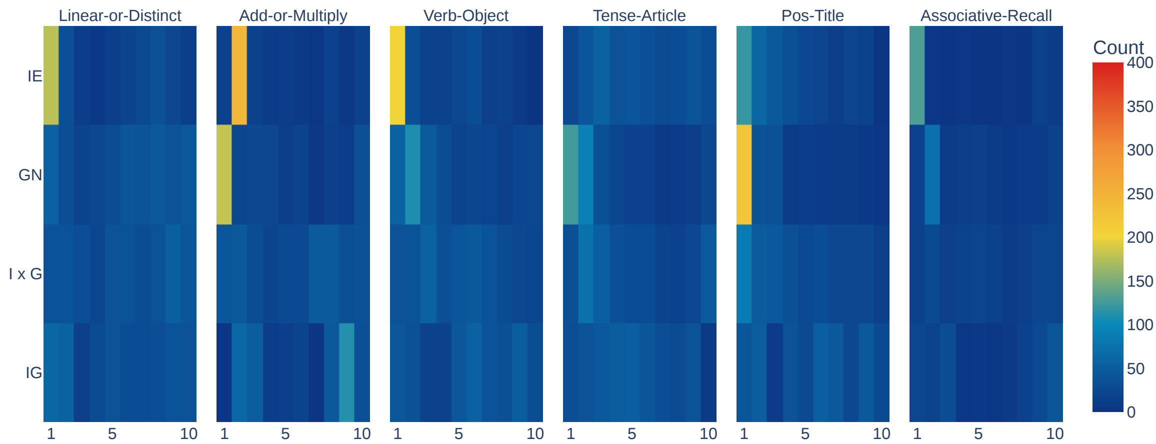

Figure 10: Distribution of the position of the example with the highest attribution scores across IA methods (Gemma-2-27B model).