Sentence-Level Explicit Inductive Inference against The Attestation Bias of LLMs

Anonymous ACL submission

Abstract

Despite the evolving reasoning ability many large language models (LLMs) have performed, they are reported to hold attestation bias in inference tasks. Instead of focusing on entailment signals between a premise and a hypothesis, LLMs are easily misled by whether the hypothesis is factual in the models' knowledge. To further study this bias and mitigate its negative effect, in this paper, we propose the sentence-level explicit inductive inference pipeline. By testing our pipeline on three NLI datasets with four mainstream LLMs, we demonstrate that although the attestation bias is still a severe problem, it can be exploited to improve LLMs' inference performance and mitigate the bias itself.[1]

1 Introduction

While many contemporary large language models (LLMs) claim to have strong reasoning ability, recent studies have shown that they are subjected to severe attestation bias in fundamental natural language inference tasks (McKenna et al., 2023). When a model is asked to predict whether a premise entails a hypothesis, instead of focusing on the inference signal, an LLM can be easily distracted by the hypothesis' out-of-context factuality. As a result, LLMs usually perform worse when the entailment label between premise and hypothesis disagrees with the attestation (factuality) label of the hypothesis.

As a solution to this, previous work has proposed the idea of explicit inductive inference. By doing inference on alternative entailment inquiries that are created by LLMs themselves, the attestation bias can be utilized to mitigate itself, and therefore improve LLMs' performance on triple-level inference tasks (Liu et al., 2024). However, in most mainstream natural language inference (NLI) challenges and downstream applications, entailment inquiries are often not presented as pairs of well-structured triples. An inference model is usually expected to pick up the entailment structure between sentences on its own, which limits the scenario where this pipeline can be used.

On the other hand, inference signals do not always lie between triples or predicates. When put into a general sentence-level inference task, whether attestation bias can still be exploited remains an open question. More and more LLMs with improved reasoning capability have been updated rapidly in recent years. This encourages us to further examine if the attestation bias is still harmful to the latest LLMs.

Based on these motivations, in this paper, we follow the idea of exploiting the attestation bias to do explicit inductive inference, and expand that methodology to present a more robust **S**entence-**L**evel **E**xplicit **I**nductive **I**nference (**SLEII**) pipeline that works on general sentence-level inference tasks. We demonstrate that this pipeline substantially improves state-of-the-art LLMs' performance on general inference tasks. Then we analysis the attestation bias hidden in latest LLMs, and show that our pipeline is also an effective solution against this bias.

2 Related work

It is widely observed that LLMs accumulate a bias towards facts that they memorized from a vast amount of pre-training corpus (Roberts et al., 2020; Carlini et al., 2022; Yan et al., 2022). In specific reasoning scenarios like solving math problems, this bias causes LLMs to perform worse even when only variable names are changed (Gulati et al., 2024).

For inference tasks, McKenna et al. (2023) designed a list of experiments specifically focused on the attestation bias, and showed that it causes LLMs to expose a significant performance drop at

[1] Our codes and data will be released upon publication.

inference time. The following works have also confirmed that this bias keeps affecting the inference performance of newer LLMs (Liu et al., 2024).

Other works proposed different ways to alleviate the negative effect of the attestation bias by presenting models with counterfactual examples (Wang et al., 2022; Zhou et al., 2023; Wang et al., 2023) or using type labels to mask the entities (Zhou et al., 2024).

For LLMs, Liu et al. (2024) proposed a novel explicit inductive inference pipeline that can utilize the attestation bias to mitigate itself. On triple-level inference tasks, they argue that if the premise can be controlled to be attested, the attestation label of the hypothesis will then statistically align with the entailment label between the premise and the hypothesis, which makes the attestation bias unharmful. Although this idea brings an interesting possibility, whether it can be applied to more complex circumstances remains an open question, which we will try to answer in this paper.

3 Sentence-Level Explicit Inductive Inference pipeline

The methodology of the SLEII pipeline is to explicitly help an LLM to expand a single entailment inquiry to more similar variant cases, and inductively draw a more reliable conclusion from all of them. Following this idea, the SLEII pipeline first leads the LLM to create a list of variations of one original entailment inquiry by replacing its semantic components. Then, the LLM will generate entailment prediction scores for these new variations as evidence to support answering the original entailment inquiry.

Each original entailment inquiry, namely a pair of one premise and one hypothesis, will go through four different modules. To better present the function of each module, we take the following pair of sentences as an example to illustrate how this pipeline works:

> Premise: John is eating chocolate after meeting Mary.
>
> Hypothesis: He received some chocolate as a gift.

Now we introduce the four modules one by one.

3.1 Alignment

This module determines which part of the sentences should be substituted. By creating variations of the original sentence pair, we aim to alter the meaning of the sentences within a reasonable range while keeping the entailment label between the sentences unchanged. To achieve this, we only replace those entities that appear in both sentences. This module locates these entities and tags them with type labels.

Type labels are necessary here to prevent ambiguity problems. While (X kills Y) entails (X is a cure of Y) when X is medicine and Y is a disease, in most of the other cases their meaning is entirely opposite. To avoid this kind of undesirable substitution, the alignment module is asked to provide a contextualized type label for each entity. Here different mentions of the same entity are allowed to have different type labels.

We encourage this module to do reference resolution between the premise and the hypothesis. A reference is counted as an entity but it may be tagged with the type labels like "pronoun". For instance, after going through this module, the example will become:

> Premise: [entity#1: person] is eating [entity#2: food] after meeting Mary.
>
> Hypothesis: [entity#1: pronoun] received some [entity#2: food] as a gift.

3.2 Premise variation

Once we obtain the tagged premise and hypothesis, we independently instantiate the premise into alternative variations. We encourage the LLM to write factual sentences if possible. These new premises are instantiated with entities that are more familiar to the LLM, and therefore these premises are more likely to be attested.

For each premise, we create k different new alternatives. Now our example looks like this:

> Premise$'_1$: Jane is eating an apple after meeting Mary.
>
> Premise$'_2$: Steve is eating popcorn after meeting Mary.
>
> .....
>
> Premise$'_k$: Tom is eating chips after meeting Mary.

An explanation of the mapping relations between tags and entities will be generated and passed to the next module.

3.3 Hypothesis instantiation

This module then derives the alternative hypotheses, based on the mapping information received from the previous module on how entities in the corresponding premise are replaced. Now we have a list of k variations of new sentence pairs. The example will look like this:

> Premise$'_1$: Jane is eating an apple after meeting Mary.
>
> Hypothesis$'_1$: She received some apples as a gift.
>
> .....
>
> Premise$'_k$: Tom is eating chips after meeting Mary.
>
> Hypothesis$'_k$: He received some chips as a gift.

3.4 Prediction

Finally, based on each alternative pair of (Premise$'_k$, Hypothesis$'_k$), this module queries the LLM to get an alternative score s_k. Note the score of the original entry as s_0, the final weighted SLEII score S_w is calculated with a weight parameter w:

$$S_w = (1 - w)s_0 + w \sum_{i=0}^{k} s_i \quad (1)$$

4 Experimental setup

4.1 Dataset

To evaluate our pipeline on sentence-level inference tasks, we test the SLEII pipeline on three NLI datasets.

SNLI The Stanford Natural Language Inference dataset (Bowman et al., 2015) is a classic NLI dataset that provides three-way classification (entailment/contradiction/neutral) entailment inquiries. It contains 570k human-written sentence pairs with crowd-sourced labels, and is widely used for NLI evaluation. Results are reported on the test set.

RTE The Recognizing Textual Entailment dataset from GLUE (Wang et al., 2019) integrate the data from the RTE challenge series. The texts are derived from real-world corpus like news and scientific articles, which makes them a suitable source of potential attestation bias. In this dataset, the "neutral" and "contradiction" labels are merged into the "not_entailment" label. Since the labels of the test set are not accessible for RTE, we report our results on the validation set.

MNLI The Multi-Genre Natural Language Inference dataset (Williams et al., 2018) is also a widely used dataset formatted the same as SNLI but with sentences from more diverse corpus like transcribed speech, fiction, and government reports. The test set's golden labels are also not directly available. We report results on the "dev_matched" development set.

For all datasets, we learn the best value of the weight parameter w on the training set under each setup and then use it to yield results on the testing sets. Due to the sheer volume of the test sets against our limited testing resources, results on the SNLI and MNLI datasets are reported for only the first 1,000 entries.[2]

4.2 Large language models

We aim to cover the latest state-of-the-art LLMs from various sources. In this paper, our pipeline is tested with 4 mainstream LLMs that are claimed to have robust reasoning abilities.

GPT-4o mini (OpenAI, 2024) is a cost-efficient variant of OpenAI's GPT-4 architecture. It claims to have powerful reasoning abilities over many natural language understanding benchmarks. The version that we use is "gpt-4o-mini-2024-07-18".

LLama 3 (Meta, 2024) is an open-source LLM published by Microsoft. In our experiment, we choose the "Llama3-8B-Instruct" version, a smaller version with 8 billion parameters.

Gemini 2.0 Flash (Google, 2024) is an enhanced comprehensive model published by Google. The version we used in our experiments is "gemini-2.0-flash".

Claude 3.5 Haiku (Anthropic, 2024) is an updated version of Anthropic's fastest LLM. The model version that we use is "claude-3-5-haiku-20241022" .

When we need to collect the probability of choices, if the LLM provides token probability access, we use the probability at the output choice mark token (A, B, or C) to represent the choice probability. When the token probabilities are not accessible, we assign 1 to the returned choice and 0 to the others.

To guarantee replicable results, whenever we query one of the LLMs, either the temperature parameter is set to 0, or the "do_sample" flag is turned off.

[2]Results on more data will be included upon publication.

4.3 Prompts

In each module, the content of the prompts may affect the final results. We tried different prompt variations in our pilot studies on the training sets, and fixed the prompts that we use before doing final experiments on the test sets.

Following Liu et al. (2024), we set the number of variations k to 10 in our experiments. For alignment, premise variation, and hypothesis instantiation module, we give few-shot examples to guarantee expected results. For the prediction and factuality determination module, we use zero-shot prompts to avoid instability from prompt engineering. The actual prompts that are used in this paper are too lengthy to include here. We present them in a separate file in our published repository.

5 Results and discussion

In this section, we test the performance of the SLEII pipeline with various experimental settings. We first illustrate an overall improvement, and then we present further analysis against the attestation bias.

5.1 Overall performance

Table 1 shows the overall performance of the SLEII pipeline. In addition to a common AUC score that indicates the area under the precision-recall curve, following McKenna et al. (2023) and Liu et al. (2024), we also calculate the normalized area-under-curve (AUC_{norm}) scores. This metric measures how well a model performs over a dummy model that predicts "entailment" to every entry. We compare our results to the baseline where we directly prompt an LLM with the original entailment inquiry.

The SLEII_{bw} marks the results when the weight parameter w in equation 1 is set to the best value learned from training sets. The pure SLEII scores are calculated by setting w to 1, which means it does not look at the original entailment inquiry at all.

It can be observed that SLEII_{bw} outperform the pure SLEII pipeline and the raw baseline under every combination of LLM and dataset. The only exception turns up when GPT4o-mini is tested on the RTE dataset. These results indicate that with a little training, the SLEII_{bw} pipeline can be used as a robust tool to improve LLMs' performance on various inference tasks.

Although in some cases, the performance of the pure SLEII pipeline is worse than the baseline, their combination SLEII_{bw} instead outperforms the baseline. This phenomenon indicates that the baseline and the SLEII pipeline make up for each other's disadvantages. By reasoning on various alternative scenarios, the combined pipeline can output more robust predictions.

Now we discuss how well does SLEII pipeline works against the attestation bias. We will only present the AUC_{norm} scores for the following sections.

5.2 Determining attestation

To better understand the analysis of attestation bias, we first inspect the distribution of the dataset in terms of attestation. For each entry, we ask the LLM to determine if the premise is factual (**F**), not factual (**NF**), or if its factuality can not be determined (**ND**). Based on the answers, we further divide each dataset into three subsets. Table 2 shows the ratio of the subsets' sizes and the AUC_{norm} results under each circumstance.

There are considerable differences between the LLMs' judgement on factuality, which is good proof of the distinction between "factuality" and "attestation". In several cases, the proportion of either the **NF** category or the **ND** category is lower than 5%. This is intuitively understandable since the boundary between "not factual" and "factuality can not be determined" is hard to define.

We do not draw any conclusion from this table. The only reason we present these AUC_{norm} scores is to set a comparison with the results in the next section. We want to emphasize that the attestation bias does not come from the attestation label itself. There is no pattern appearing when we divide the data according to their attestation category. The damage that the attestation bias causes can only be observed if we focus on the relation between the attestation labels and the entailment labels.

With this preparation, we now present the analysis of the attestation bias.

5.3 Against the attestation bias

Following McKenna et al. (2023) and Liu et al. (2024), under each setting, we divide the dataset into the attestation-consistent subset (*cons.*) and the attestation-adversarial (*adv.*) subsets, according to whether the entailment label of one entry is the same as the factuality label of its hypothesis. Here we only accept the factual and not factual hypotheses, but discard any entry if the factuality

Model	Pipeline	Dataset					
		SNLI		RTE		MNLI	
		AUC	AUC_{norm}	AUC	AUC_{norm}	AUC	AUC_{norm}
Llama3-8B	-	85.8	78.4	89.9	78.8	76.2	63.9
	SLEII	68.1	51.5	83.9	66.2	71.8	57.3
	SLEII_{bw}	**86.5**	**79.5**(w=0.12)	**90.4**	**79.8**(w=0.01)	**76.3**	**64.2**(w=0.23)
GPT4o-mini	-	96.0	93.9	95.9	91.5	89.8	84.5
	SLEII	89.5	84.0	91.3	81.7	88.1	81.9
	SLEII_{bw}	**96.3**	**94.4**(w=0.04)	**95.9**	**91.5**(w=0)	**90.0**	**84.8**(w=0.36)
Gemini-2.0	-	89.8	84.4	88.5	75.8	83.1	74.3
	SLEII	91.9	87.7	89.9	78.7	85.2	77.5
	SLEII_{bw}	**92.6**	**88.8**(w=0.28)	**90.8**	**80.6**(w=0.54)	**85.5**	**78.0**(w=0.46)
Claude-3.5	-	86.4	79.4	87.4	73.5	83.8	72.2
	SLEII	87.3	80.7	87.3	73.2	86.7	77.3
	SLEII_{bw}	**89.7**	**84.4**(w=0.44)	**88.7**	**76.1**(w=0.61)	**87.5**	**78.6**(w=0.57)

Table 1: The overall pipeline performance under each setup in terms of the area under the precision-recall curve (AUC) and the normalized area-under-curve (AUC_{norm}). SLEII_{bw} marks the results using the best w value learned from the training set under the same setting. The best results under each setting are highlighted.

Model	Pipeline	Dataset								
		SNLI			RTE			MNLI		
		F	NF	ND	F	NF	ND	F	NF	ND
	(data ratio)	*83%*	*16%*	*0%*	*91%*	*8%*	*0%*	*56%*	*42%*	*1%*
Llama3-8B	-	78.5	85.8	-	79.2	70.4	-	71.4	56.7	-
	SLEII	48.9	76.3	-	66.9	55.4	-	63.4	52.6	-
	SLEII_{bw}	79.3	84.8	-	80.4	70.4	-	71.2	57.5	-
		5%	*90%*	*4%*	*17%*	*58%*	*25%*	*39%*	*42%*	*18%*
GPT4o-mini	-	91.7	94.4	-	90.3	91.0	92.7	80.5	89.1	85.5
	SLEII	83.6	84.4	-	82.1	82.1	83.8	81.1	83.4	83.3
	SLEII_{bw}	90.5	95.1	-	90.3	91.0	92.7	83.8	87.6	83.6
		20%	*1%*	*80%*	*65%*	*22%*	*12%*	*31%*	*13%*	*54%*
Gemini-2.0	-	80.1	-	85.8	76.5	74.3	75.0	73.1	72.5	75.6
	SLEII	85.8	-	88.2	80.4	72.7	78.1	75.5	78.5	78.7
	SLEII_{bw}	87.5	-	89.0	81.8	77.4	78.6	76.3	78.5	79.3
		47%	*1%*	*51%*	*61%*	*13%*	*36%*	*24%*	*4%*	*72%*
Claude-3.5	-	77.8	-	80.8	72.6	77.0	77.6	78.9	-	70.5
	SLEII	80.1	-	81.2	70.5	75.0	78.3	80.6	-	77.3
	SLEII_{bw}	84.1	-	84.7	75.2	77.4	82.0	84.9	-	78.1

Table 2: Conditional pipeline performance when the premise is determined by the LLM to be factual (F), not factual (NF), or its factuality can not be determined (ND). All results are measured by the normalized area under the precision-recall curve (%).

Model	Pipeline	Dataset								
		SNLI			RTE			MNLI		
		cons.	*adv.*	***diff.***	*cons.*	*adv.*	***diff.***	*cons.*	*adv.*	***diff.***
Llama3-8B	-	96.9	23.6	**-73.3**	94.1	39.8	**-54.3**	83.1	30.0	**-53.1**
	SLEII	70.6	11.9	-58.7	78.5	31.8	-46.7	76.0	25.9	-50.1
	SLEII$_{bw}$	95.5	29.2	-66.3	92.8	46.0	-46.8	82.0	31.2	-50.8
GPT4o-mini	-	65.9	97.6	+31.7	75.8	96.9	**+21.1**	77.7	88.8	+11.1
	SLEII	35.7	92.2	+56.5	54.1	93.3	+39.2	74.5	86.4	+11.9
	SLEII$_{bw}$	71.8	97.5	**+25.7**	75.8	96.9	**+21.1**	78.4	88.9	**+10.5**
Gemini-2.0	-	99.1	51.7	**-47.4**	85.6	67.0	**-18.6**	78.9	65.9	**-13.0**
	SLEII	98.9	54.1	-44.8	82.4	81.9	-0.5	80.8	71.2	-9.6
	SLEII$_{bw}$	99.0	54.6	-44.4	85.1	82.0	-3.1	80.8	71.7	-9.1
Claude-3.5	-	95.3	50.4	**-44.9**	95.3	40.0	**-55.3**	92.1	68.2	**-23.9**
	SLEII	91.1	48.2	-42.9	89.8	46.5	-43.3	87.8	73.6	-14.2
	SLEII$_{bw}$	94.5	50.1	-44.4	94.4	43.6	-50.8	93.4	72.4	-21.0

Table 3: AUC$_{norm}$ (%) scores on attestation-consistent (cons.) and attestation-adversarial (adv.) subsets. The diff. column marks the difference from cons. to adv. The lowest diff. value under each setting is highlighted.

of their hypothesis can not be determined. For entailment labels, the "neutral" and "contradiction" labels are all treated as "not_entailment" labels. In addition, we add the ***diff.*** columns to display the performance drop between the *cons.* and *adv.* columns.

Table 3 shows the results on these two kinds of subsets. Under each setting, the lowest value in the ***diff.*** columns is in boldface. To avoid misleading the LLMs, we discard the "If *premise* then *hypothesis*, is that true?" prompt that McKenna et al. (2023) and Liu et al. (2024) used, which were suspected to be one reason why the LLMs are distracted by factuality. However, even when our baseline results have excluded that factor, the negative values in the ***diff.*** columns are still significantly high.

For Llama, Gemini and Claude, a huge performance drop shows up in the baseline setting, which exposes the severe attestation bias these SOTA LLMs hold within their reasoning mechanism. While new generations of LLMs are achieving higher scores at various benchmarks constantly, their inference inability caused by the attestation bias has still been proved hard to resolve. We hope these results can raise more attention from the LLMs evaluation community.

GPT4o-mini appears to be an interesting exception in this experiment, as it surprisingly shows a reversal performance improvement against the attestation bias. That means this model performs even better when the entailment label contradicts the attestation label of one entry. More curiously, the SLEII pipeline can still alleviate this bias towards zero, but now in the opposite direction. Due to the black-box nature of this model's training process, we can not yet offer a convincing explanation for this.

For the other LLMs, the baseline always yields worse performance drops compared to using the SLEII pipeline. This comparison supports the conclusion that the attestation bias can be exploited to mitigate itself, even in general sentence-level inference cases. By applying the SLEII pipeline, an LLM can now give a more accurate prediction when the entailment inquiries are against the attestation bias.

6 Conclusion

In this paper, we apply the idea of explicit inductive inference to sentence-level inference tasks and propose the SLEII pipeline. By demonstrating the performance of our pipeline with four latest LLMs on three typical NLI datasets, we make two points in summary:

The attestation bias is still a severe problem existing within many SOTA LLMs. In general inference tasks, this bias substantially undermines LLMs' inference performance and restricts their reasoning ability in a way that is hard to notice. We

call for more attention to both further studying this bias and finding more solutions to this challenge.

The sentence-level explicit inductive inference pipeline works on general inference tasks. By creating variations of one entailment inquiry as extra evidence and making inferences on them, an LLM can improve its general reasoning performance, and output more robust entailment prediction against the attestation bias.

Limitations

For some modules in this paper, it is possible that prompt engineering may substantially affect the outcome. For example, the variation that appeared in section 5.2 may possibly be controlled by adding few-shot examples in the input prompt. These may require future work.

Using the SLEII pipeline to do inference on k extra variations results in k times of extra resource spent. This method may not be suitable for cases where a model needs to process a large amount of entries.

References

Anthropic. 2024. Claude 3.5.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Google. 2024. Gemini 2.0.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. 2024. Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Tianyang Liu, Tianyi Li, Liang Cheng, and Mark Steedman. 2024. Explicit inductive inference using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15779–15786, Miami, Florida, USA. Association for Computational Linguistics.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

Meta. 2024. Llama3.

OpenAI. 2024. Gpt 4o-mini.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15173–15184, Singapore. Association for Computational Linguistics.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3071–3081, Seattle, United States. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. 2022. On the robustness of reading comprehension models to entity renaming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 508–520, Seattle, United States. Association for Computational Linguistics.

Ben Zhou, Hongming Zhang, Sihao Chen, Dian Yu, Hongwei Wang, Baolin Peng, Dan Roth, and Dong Yu. 2024. Conceptual and unbiased reasoning in language models. *arXiv preprint arXiv:2404.00205*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

A Computational cost

Our experiments on Llama3-8B-Instruct are applied on one A6000 GPU. For every 1,000 entries (10 variations for each entry), the average time of running through the entire pipeline is 6 hours.

Other experiments are executed with online APIs. Typical consumed time for every 1,000 entries: 9 hours for GPT 4o-mini, 6 hours for Gemini 2.0 Flash, 21 hours for Claude 3.5 Haiku.