Explicit Inductive Inference using Large Language Models

Tianyang Liu **Tianyi Li** **Liang Cheng** **Mark Steedman**
University of Edinburgh
T.Liu-47@sms.ed.ac.uk tianyi.li@ed.ac.uk
L.Cheng-13@sms.ed.ac.uk m.steedman@ed.ac.uk

Abstract

Large Language Models (LLMs) are reported to hold undesirable attestation bias on inference tasks: when asked to predict if a premise P entails a hypothesis H, instead of considering H's conditional truthfulness entailed by P, LLMs tend to use the out-of-context truth label of H as a fragile proxy. In this paper, we propose a pipeline that exploits this bias to do explicit inductive inference. Our pipeline uses an LLM to transform a premise into a set of attested alternatives, and then aggregate answers of the derived new entailment inquiries to support the original inference prediction. On a directional predicate entailment benchmark, we demonstrate that by applying this simple pipeline, we can improve the overall performance of LLMs on inference and substantially alleviate the impact of their attestation bias.[1]

1 Introduction

Large Language Models (LLMs) are claimed to possess *implicit* inductive reasoning ability through pre-training: from the massive examples they memorized, they draw inference rules and encode them latently so that they can apply these rules to do reasoning at test time.

However, recently McKenna et al. (2023a) has pointed out that LLMs are severely affected by an attestation bias when performing inference tasks. Given the question of whether premise P entails hypothesis H with few-shot examples, an LLM's prediction is deeply bound to the hypothesis' out-of-context truthfulness, instead of its conditional truthfulness entailed by the premise. When the hypothesis H is attested in an LLM's world knowledge (the LLM believes H to be true), the LLM is likely to predict the entailment to be true, regardless of the premise. As a result, LLMs suffer a significant performance drop when the entailment labels disagree with the attestation label of hypothesis H.

[1]Codes and data of this paper are available at https://github.com/waterltyang/EIDI

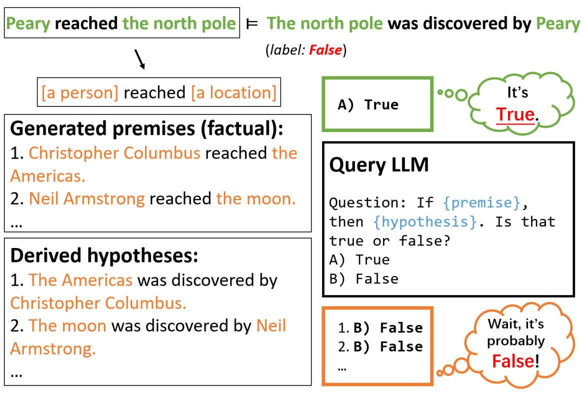
Peary reached the north pole ⊨ The north pole was discovered by Peary
(label: False)
[a person] reached [a location]
Generated premises (factual):
1. Christopher Columbus reached the Americas.
2. Neil Armstrong reached the moon.
...
Derived hypotheses:
1. The Americas was discovered by Christopher Columbus.
2. The moon was discovered by Neil Armstrong.
...
A) True
It's True.
Query LLM
Question: If {premise}, then {hypothesis}. Is that true or false?
A) True
B) False
1. B) False
2. B) False
...
Wait, it's probably False!

Figure 1: An example of the explicit inductive inference pipeline. While direct entailment inquiry gets a wrong answer, it can be corrected by reasoning on more alternative examples.

Although this is a severe problem limiting LLMs' performance on non-attested inferences, we argue that with careful design, this bias can instead be **exploited** to improve LLM performance on inference tasks. We notice a statistically true conclusion: Given an entailment inquiry $P \models H$, the attestation bias is harmful only when the premise P is not attested. If we control P to always be attested, then $P \models H$ will naturally share the same truth label with H on a distributional basis, which dissolves the negative effects of LLMs' attestation bias.

Applying this idea, we propose a simple yet effective Explicit Inductive Inference pipeline with LLMs. As illustrated in Figure 1, the core idea is to transform a premise into a set of attested alternatives by replacing the arguments, and to aggregate the LLM's predictions on these derived inquiries to support answering the original question.

We test our pipeline with two LLMs on Levy/Holt (Levy and Dagan, 2016; Holt, 2019), a difficult directional predicate inference dataset, and further analyze the influence of our pipeline against the models' attestation bias. The results show that our pipeline can improve not only LLM's overall performance on predicate inference, but also their robustness against the attestation bias.

To summarize our contribution, we propose an easy-to-use inference pipeline that **1)** improves LLMs' performance on predicate inference, **2)** substantially alleviates negative effects of the LLMs' attestation bias, and **3)** uses LLMs' own generation capability without requiring external knowledge.

2 Related Work

LLMs accumulate a bias towards factual knowledge by encoding massive facts during pre-training (Roberts et al., 2020; Carlini et al., 2022; Yan et al., 2022). Recently, McKenna et al. (2023a) pointed out that LLMs suffer from an attestation bias on inference tasks as a result. Note that the effect of attestation bias is similar to that of the hypothesis-only baseline (Poliak et al., 2018), but while the former is a bias from pre-training, the latter originates from dataset artifacts in supervised learning.

In other tasks, previous work has mitigated the bias towards attestation by introducing counterfactual examples (Wang et al., 2022b; Zhou et al., 2023; Wang et al., 2023) or replacing argument entities with their type labels (Zhou et al., 2024). In this paper, we go one step further to show that in an inference task, we should instead encourage the models to generate factual alternative examples.

The idea of aggregating multiple versions of LLMs' outputs has been explored in prior work. Wang et al. (2022a) encourage LLMs to generate multiple reasoning paths for one question, while Zhou et al. (2022) embody one question with multiple prompts. In contrast, our method creates semantically different alternative questions, which serve as extra evidence for an original inquiry.

3 Explicit Inductive Inference

3.1 Task and Definition

The task of this work is to determine the entailment relation between two binary predicates where both predicates are contextualized with the same pair of entities. The input will be in the form of two triples $(s, p, o) - (s, h, o)$ where s is the subject entity, o is the object entity, p is the premise predicate, and h is the hypothesis predicate. There are also cases in the form of $(s, p, o) - (o, h, s)$ where the two entities are swapped in position like the example in Figure 1. Without loss of generality, we describe our method with inputs in the former format.

The goal is to predict whether the premise triple entails the hypothesis triple, namely the truth label of $(s, p, o) \models (s, h, o)$. To use an LLM to predict entailments, each input triple pair will be wrapped into a prompt. We mark them as $Q[(s, p, o) \models (s, h, o)]$ and call them entailment inquiries.

3.2 Exploit the Attestation Bias

As stated in Section 1, the attestation bias of LLMs can be less detrimental if the premise P is attested in an entailment inquiry, because the truth label of $P \models H$ would likely be the same as the attestation label of H. Besides this, two more insights are guiding our pipeline design:

1) The label of a predicate entailment inquiry does not change when the argument entities are replaced, as long as the substitution entities keep the same semantic type labels.

2) Factual $\neq$ Attested. Factual knowledge from external sources may not be confirmed by LLMs for being longtail, absent in pre-training data, or conflicted with out-of-date records. Facts generated by LLMs, on the other hand, are highly likely to be recognizable by themselves. Even hallucinated generations are acceptable since they are still attested if not factual.

Based on these understandings, we propose the **E**xplicit **InD**uctive **I**nference (EIDI) pipeline. Given an entailment inquiry $P \models H$, our EIDI pipeline first transforms P into a set of different attested premises P's by replacing the arguments of P. Then the corresponding set of $H's$ is derived, so that we now have a list of alternative inquiries $P' \models H'$. Finally, we explicitly do an inductive inference on these new inquiries by drawing a concluded entailment prediction from an LLM's answers to these alternative inquiries.

It is worth mentioning that given P is true, logically, H is always true if $P \models H$ but not vice versa. We can only statistically conjecture $P \models H$ if we observe a high probability of H being true (predicted by the LLM according to the bias). Therefore, we encourage the transformation module to generate a variety of different alternative premise triples, so that a more reliable conclusion can be drawn when we aggregate the predictions.

3.3 Explicit Inductive Inference Pipeline

Typing While the label of (medicine X, *kills*, disease Y) $\models$ (medicine X, *is a cure of*, disease Y) is *True*, one can not therefore deduce that (Person X, *kills*, animal Y) $\models$ (Person X, *is a cure of*, Animal

Y). To prevent these errors incited by the ambiguity of predicates, for each premise triple (s, p, o), we query the LLM to obtain the entity type label of the subject and object t_s and t_o. Here we do not predefine a vocabulary for possible type labels since the purpose is only to disambiguate.

Transformation With these assigned type labels we query the LLM to generate alternative arguments for the premise predicate. From one typed premise triple (s, t_s, p, o, t_o), we encourage the LLM to generate a list of new attested triples $(s_1, p, o_1), ..., (s_n, p, o_n)$ where the substitution entities keep the original types, i.e. any s_i still has type t_s and any o_i still has type t_o. These n new premise triples will then be expanded to n new entailment inqueries $Q[(s_1, p, o_1) \models (s_1, h, o_1)], ..., Q[(s_n, p, o_n) \models (s_n, h, o_n)]$.

Prediction At this point, we input each derived entailment inquiry $Q[(s_i, p, o_i) \models (s_i, h, o_i)]$ to the LLM to get their response and corresponding probability score. Then we take the average score of these n different scores as our explicit inductive score for the original entailment inquiry.

4 Experimental Setup

4.1 Datasets

We test our pipeline on the Levy/Holt dataset (Levy and Dagan, 2016; Holt, 2019), a predicate entailment dataset where each entry consists of two triples in the form of $(s, p, o) - (s, h, o)$ or $(s, p, o) - (o, h, s)$, and a following label shows whether the premise triple entails the hypothesis triple. We use the directional portion of this dataset following prior work (McKenna et al., 2023b; Chen et al., 2022; Li et al., 2022), as it is a challenging test focused on the understanding of entailment beyond bi-directional similarity.

Following McKenna et al. (2023a), we further analyze how the LLMs' attestation bias is digested in our method. We split the Levy/Holt dataset according to whether the label of $P \models H$ agrees with the attestation label (obtained by querying the LLM) of H for each entry. For 1784 entries in the directional test set, this yields a 956 : 828 split of attestation-consistent : attestation-adversarial subsets from querying GPT-3.5, and a similar 997 : 787 split with Llama3.

The substantial size of the attestation-adversarial subset demonstrates the detrimental effect of attestation bias in real datasets. We report results on both the directional test set and its two subsets in Section 5.

4.2 LLMs

We test our method with two LLMs, GPT-3.5 and Llama3. GPT-3.5 (OpenAI, 2023) is a set of powerful closed-source commercial LLMs. We choose the GPT-3.5-Turbo version for its widespread use in the research community. Llama3 (Meta, 2024) is a SOTA open-source LLM, where we choose the largest Llama3-70B-instruct version for its optimized capacity. Throughout our experiments, we use greedy decoding for reproducible results.

Our pilot studies on the development set indicate that adding few-shot examples in the prediction module may add extra bias to the model, and therefore introduce unnecessary considerations on finding proper examples under each setting. Hence we choose zero-shot prompts for the prediction module and one-shot prompts for the transformation module where the only example is the original premise. Discussion on prompt selection and a list of all prompts we use are included in Appendix A.

4.3 Baselines and Metric

We compare EIDI against two baselines. We construct MCQ$_{entity}$ baseline by directly wrapping the original premise and hypothesis with the Multipe-Choice Question prompt used in our prediction module, and passing it to the LLM to get an entailment prediction. MCQ$_{type}$ baseline is set up in the same way where the only difference is that we first replace the arguments of the predicates by their entity types. To keep ourselves aligned with previous work, we use the 48 FIGER types (Ling and Weld, 2012) as in McKenna et al. (2023a) for this measure, instead of the LLM-generated types in Section 3.3.

We draw the precision-recall curve for EIDI and each baseline by inspecting the final output token probability of the model's response. As a result of the multiple-choice prompt design, returned answers always start with a choice mark where A is for entailment and B is for non-entailment. For baseline methods, we score that one token's probability. For our EIDI pipeline, we calculate the average score of the k output tokens' probabilities.

Following Li et al. (2022); McKenna et al. (2023a), we calculate the normalized area-under-curve (AUC$_{norm}$) as an indicator of the model's performance. This measure describes how much

	Model	
Pipeline	GPT-3.5	Llama3
MCQ_{entity}	23.85	36.66
MCQ_{type}	25.88	35.13
EIDI_{all}	**35.52**	**50.89**
EIDI_1	31.16	41.85
EIDI_2	32.10	46.75
EIDI_5	33.41	49.61

Table 1: Overall normalized Area-Under-the-Curve (%) of our EIDI pipeline and the two baselines on the full Levy/Holt directional test set. EIDI_i inspects only i alternative inquiries, and EIDI_{all} considers all examples obtained in the transformation step.

Model	**Pipeline**	*cons.*	*adv.*	***diff.***
GPT-3.5	MCQ_{entity}	82.04	0.00	*-82.04*
	MCQ_{type}	69.40	0.48	*-68.92*
	EIDI_{all}	56.14	9.97	***-46.17***
	EIDI_1	53.73	8.95	***-44.78***
Llama3	MCQ_{entity}	81.08	0.01	*-81.07*
	MCQ_{type}	70.25	2.41	*-67.84*
	EIDI_{all}	69.59	23.83	***-45.76***
	EIDI_1	63.98	15.66	*-48.32*

Table 2: AUC_{norm} (%) on the attestation-bias-split datasets. The ***diff.*** column marks the difference between results on the attestation-consistent (*cons.*) and attestation-adversarial (*adv.*) subsets.

better a model is over a degenerate baseline returning positive answers to every data entry.

5 Results and Discussion

5.1 Overall performance

Table 1 shows the performance of each model on the directional Levy/Holt test set. With both LLMs, our EIDI_{all} pipeline gains a significant improvement over the two baseline methods.

The typical value of the size of total generated examples n is 10 for the EIDI_{all} setting. It can be observed that the performance of EIDI_i steadily increases along with i, confirming our hypothesis that with attested P's, the more cases of alternative $P' \models H'$ generated, the more reliable our pipeline is. The complete results of all EIDI_i settings are shown in Appendix B.

An interesting observation lies between the performance of the EIDI_1 setting and the baselines, which shows that replacing the original inquiry with even one self-generated example can improve the LLMs' predicate inference performance. The difference between EIDI_1 and MCQ_{type} baseline also highlights the importance of instantiating attested triples. Since the effect of the attestation bias is already excluded from the results of the MCQ_{type}, this proves that the EIDI pipeline is taking advantage of further exploiting the bias.

5.2 Against the bias

Table 2 compares the performance of each method on attestation-consistent (*cons.*) and attestation-adversarial (*adv.*) subsets. Measured by the difference of AUC_{norm} between the two subsets, our pipeline reduces the effect of LLMs' attestation bias by over 20% from the MCQ_{type} baseline, and over 35% from the MCQ_{entity} baseline in average.

With both LLMs, we observe an AUC_{norm} of near 0% in the two baseline settings, demonstrating the extreme inability of the LLMs to capture the essential entailment signal against the attestation bias in a zero-shot setting.

Interesting results appear again under the EIDI_1 setting. On GPT-3.5-turbo, it slightly outperforms the EIDI_{all} setting. But this only happens because EIDI_{all} setting is doing better on the attestation-consistent subset, which implies that EIDI_{all} setting is still the choice for best performance, while EIDI_1 is also a strong candidate for scenarios with limited compute.

These results suggest that our pipeline can be used to improve LLMs' general inference performance, and especially in attestation-adversarial scenarios, e.g. *If lions are fed on hay, then lions eat hay.* As a replacement to LLM's direct inference prediction, EIDI pipeline can be easily plugged into frameworks with LLMs to do various downstream tasks like question answering and KG completion.

6 Conclusions

We propose an explicit inductive pipeline exploiting the attestation bias of LLMs to do more robust predicate inference. With experiments on the directional Levy/Holt dataset and its attestation-bias-split subsets, we have shown that our baseline gains a significant improvement over LLM's primary inference performance, and substantially reduces the performance loss caused by LLMs' attestation bias.

Without external knowledge, EIDI use LLMs' own generation to exploit their attestation bias. Our results suggest that although biases of LLMs are usually undesirable obstacles, in some scenarios they may be tapped for good with careful design. We advocate for similar ideas to be applied to other tasks to improve LLM performance in future work.

7 Limitations

In this paper, we demonstrated the performance of our pipeline by comparing it to two baselines. Although we intend to exclude prompt engineering factors from our analysis, it has been widely accepted that including few-shot examples and other prompting techniques can guide LLMs to output better answers. Therefore there could be further studies on evaluating the effects of using different prompts in the EIDI pipeline.

Generating alternative inquiries and respectively doing inferences on them can be computationally expensive compared to only one determination in baseline settings. As a result, downstream applications may find a trade-off between computational efficiency and better inference performance.

We also tested our pipeline against the frequency bias that McKenna et al. (2023a) pointed out, and the results show that the EIDI pipeline amplifies this bias compared to the baselines due to its choice of popular entities. We argue that this reaffirms the challenge in achieving Pareto improvements on LLM robustness against biases, and leave those results and discussions to Appendix C.

8 Acknowledgement

We would like to thank the reviewers for their valuable comments and suggestions. This work was partly funded by a Mozilla PhD scholarship at Informatics Graduate School and by the University of Edinburgh Huawei Laboratory.

References

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*.

Zhibin Chen, Yansong Feng, and Dongyan Zhao. 2022. Entailment graph learning with textual entailment and soft transitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5899–5910, Dublin, Ireland. Association for Computational Linguistics.

Xavier Holt. 2019. Probabilistic models of relational implication. *Preprint*, arXiv:1907.12048.

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Tianyi Li, Mohammad Javad Hosseini, Sabine Weber, and Mark Steedman. 2022. Language models are poor learners of directional inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 903–921, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiao Ling and Daniel Weld. 2012. Fine-grained entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 94–100.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023a. 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.

Nick McKenna, Tianyi Li, Mark Johnson, and Mark Steedman. 2023b. Smoothing entailment graphs with language models. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 551–563, Nusa Dua, Bali. Association for Computational Linguistics.

Meta. 2024. Llama3.

OpenAI. 2023. Openai.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

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.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022b. 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.

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*.

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Prompt consistency for zero-shot task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2613–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 Prompts Selection

Here we list all the prompts that we use in our experiments.

Typing The purpose of this module is only to disambiguate the predicates, therefore no vocabulary of allowed type labels is predefined.

> Type the entities in the following triples:
>
> Hitler | was born in | Poland -> a person | was born in | a country
>
> Hogs | eats | Corn -> an animal | eats | a food
>
> Aspirin | may reduce the risk of | Cancer -> a medicine | may reduce the risk of | a disease
>
> $\{s\}$ | $\{p\}$ | $\{o\}$ ->

Transformation Although we use the word 'fact', the generated triples are always attested rather than factual.

> Write $\{n + 1\}$ facts in the form of " $\{t_s\}$ | $\{p\}$ | $\{t_o\}$."
>
> - $\{s\}$ | $\{p\}$ | $\{o\}$.
>
> -

Prediction This is also used for the two baselines.

> Question:If $\{s\}$ $\{p\}$ $\{o\}$, then $\{s\}$ $\{h\}$ $\{o\}$. Is that true or false?
>
> Choices:
>
> A) True
>
> B) False
>
> Answer:

For prediction module, when an instruction is required, we use the following one:

> Only return one mark A, B or C to answer the question.

B Results on all EIDI$_i$ Settings

Table 3 shows the performance of all EIDI$_i$ settings. Best performences are reached when all transformed alternative inquiries are considered.

	Model	
Pipeline	GPT-3.5	Llama3
MCQ$_{entity}$	23.85	36.66
MCQ$_{type}$	25.88	35.13
EIDI$_1$	31.16	41.85
EIDI$_2$	32.10	46.75
EIDI$_3$	31.47	47.52
EIDI$_4$	32.05	48.60
EIDI$_5$	33.54	49.61
EIDI$_6$	33.41	50.42
EIDI$_7$	34.68	50.13
EIDI$_8$	34.76	50.36
EIDI$_9$	35.28	50.39
EIDI$_{10}$	**35.52**	50.01
EIDI$_{11}$	-	50.52
EIDI$_{12}$	-	**50.89**

Table 3: AUC$_{norm}$ (%) of all EIDI$_i$ settings.

C Frequency Bias

We also tested our pipeline on the frequency bias using the same dataset split measure as that for attestation bias. The dataset that we use is from McKenna et al. (2023a)'s work, where we have 972 entries of frequency-consistent input and 220 entries of frequency-adversarial input.

Compared to baselines, the EIDI pipeline introduces extra frequency bias. This is expected since our transformation module is not designed to alter the relative frequency of the predicates, and may have amplified the frequency bias by taking popular alternative entities generated by the LLMs. This result reaffirms the challenging nature of directional inference and the difficulty in improving robustness against multiple biases at once.

Model	**Pipeline**	*cons.*	*adv.*	***diff.***
GPT-3.5	MCQ$_{entity}$	20.58	29.38	+8.80
	MCQ$_{type}$	24.49	32.93	+8.44
	EIDI$_{all}$	40.66	20.83	-19.83
	EIDI$_1$	33.94	18.83	-15.11
Llama3	MCQ$_{entity}$	33.30	47.87	+14.57
	MCQ$_{type}$	31.47	47.19	+15.72
	EIDI$_{all}$	51.97	42.27	-9.70
	EIDI$_1$	39.78	35.32	-4.46

Table 4: Normalized area-under-curve(%) on the frequency-bias-split datasets.

D Computational Cost

Our experiments on Llama3-70B-Instruct are applied on two A6000 GPUs. For 1784 entries and 10 alternative inquiries for each entry, the typing module takes about 3 GPU hour, the transformation module takes about 100 GPU hours, and the prediction module takes about 6 GPU hours.