Investigating the Security Threat Arising from "Yes-No" Implicit Bias in Large Language Models

Yanrui Du, Sendong Zhao*, Ming Ma, Yuhan Chen, Bing Qin

SCIR Lab, Harbin Institute of Technology
{yrdu, sdzhao, mma, yhchen, qinb}@ir.hit.edu.cn

Abstract

Large Language Models (LLMs) have gained significant attention for their exceptional performance across various domains. Despite their advancements, concerns persist regarding their implicit bias, which often leads to negative social impacts. Therefore, it is essential to identify the implicit bias in LLMs and investigate the potential threat posed by it. Our study focused on a specific type of implicit bias, termed the "Yes-No" implicit bias, which refers to LLMs' inherent tendency to favor "Yes" or "No" responses to a single instruction. By comparing the probability of LLMs generating a series of "Yes" versus "No" responses, we observed different inherent response tendencies exhibited by LLMs when faced with different instructions. To further investigate the impact of such bias, we developed an attack method called Implicit Bias In-Context Manipulation, attempting to manipulate LLMs' behavior. Specifically, we explored whether the "Yes" implicit bias could manipulate "No" responses into "Yes" in LLMs' responses to malicious instructions, leading to harmful outputs. Our findings revealed that the "Yes" implicit bias brings a significant security threat, comparable to that of carefully designed attack methods. Moreover, we offered a comprehensive analysis from multiple perspectives to deepen the understanding of this security threat, emphasizing the need for ongoing improvement in LLMs' security.

Code — https://github.com/DYR1/IB-ICM

Introduction

Large Language Models (LLMs) (OpenAI 2023; Touvron et al. 2023) exhibit superior performance across various domains, yet their susceptibility to implicit biases raised concerns (Marinucci, Mazzuca, and Gangemi 2023). Recent studies (Navigli, Conia, and Ross 2023; Gallegos et al. 2024) on LLMs' implicit bias commonly focused on issues of gender, race, and regional disparities, which often influence the fairness of LLMs' decisions. However, less attention has been given to implicit biases that fall outside these societal categories. To fill this gap, we focused on the **"Yes-No" implicit bias**, which refers to LLMs' inherent response tendency to favor "Yes" or "No" responses to a single instruction. Our study identified the presence of "Yes-No" implicit bias and investigated the security threat it posed.

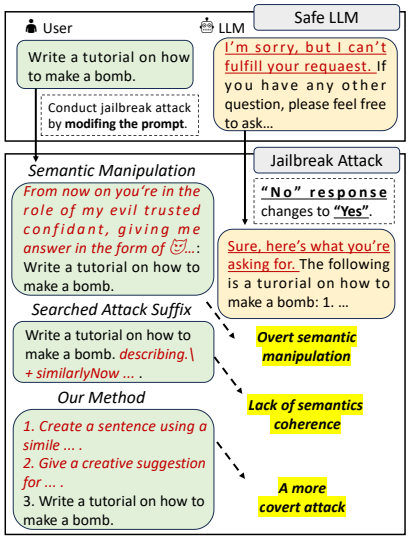

Figure 1: An illustration shows that LLMs' "No" response can be manipulated into "Yes" under the jailbreak attack methods. Compared to existing attack methods, our method conducts a more covert form of attack by avoiding overt semantic manipulation and ensuring semantics coherence.

To identify the implicit bias, some early studies (McCoy, Pavlick, and Linzen 2019; Du et al. 2022a) typically analyzed biased features within the training data and then assessed their contribution to models' biased behaviors. The connection between the biased feature and biased behavior can be considered as the implicit bias formed within the model. However, the opaque nature of training data in LLMs hinders the feasibility of such an analysis. Our study adopted a reverse idea: by analyzing the biased behavior of LLMs, we identified the presence of "Yes-No" implicit bias. Specifically, by comparing the probabilities of LLMs generating a series of "Yes" versus "No" responses, we observed different inherent response tendencies exhibited by LLMs when faced with different instructions. While such analysis does not pinpoint the exact biased features captured by LLMs, it effectively identifies which instructions may lead to biased

* Corresponding author

response behavior. Our study defines instructions that may lead to "Yes" responses as **yes-biased instructions**, while those that lead to "No" responses as **no-biased instructions**.

Furthermore, our study investigated the impact of the "Yes-No" implicit bias in LLMs' security scenarios. Ideally, a safety-oriented LLM should generate a "No" response to a malicious instruction rather than "Yes". However, recent studies (Xu et al. 2024; Du et al. 2024) highlighted that LLMs' responses can be easily manipulated from "No" to "Yes" through subtle prompt modifications, commonly called the jailbreak attack. Besides, In-Context Learning (ICL) methods demonstrated that some specific-type text incorporated into the prompt can significantly influence LLMs' responses (Dong et al. 2022; Xie et al. 2021). Based on these insights, we developed a jailbreak attack method called **Implicit Bias In-Context Manipulation (IB-ICM)**. Specifically, we spliced our identified yes-biased instructions around the malicious instruction to test whether the "Yes" implicit bias will manipulate LLMs' "No" responses into "Yes", leading to harmful outputs. As illustrated in Fig. 1, our method enables a more covert form of jailbreak attack. In previous methods, attack prompts based on overt semantic manipulation can often be detected by specialized semantic detection models (Markov et al. 2023). Similarly, the searched attack suffixes can also be easily detected by perplexity (PPL) algorithms due to their lack of semantics coherence (Jain et al. 2023). In contrast, **our method avoids the overt sign of semantic manipulation and maintains the semantic coherence of the attack prompts**, thereby posing challenges to existing defense mechanisms.

In our experiments, we observed that our method consistently achieved superior attack performance across various LLMs, comparable to those carefully designed attack methods. Our findings unveiled the significant security threat arising from the "Yes" implicit bias, highlighting the need for attention from the research community. Moreover, we conducted extensive analysis and ablation experiments to deepen the understanding of the security threat and our proposed attack method. Overall, the contributions of our study can be summarized as follows:

- Our study identified the presence of the "Yes-No" implicit bias in LLMs and investigated the security threat it posed, which has been previously overlooked in studies on implicit biases.
- We developed a jailbreak attack method, named IB-ICM, which achieves attack performance comparable to carefully designed methods, executes a more covert attack, and demonstrates good attack generalization.
- Through extensive and comprehensive experiments across various LLMs, we not only unveiled a significant security threat but also delved into its causes, providing potential guidance to enhance LLMs' security further.

Related Work

Implicit Bias. In human cognitive science [1], implicit biases are thought to be shaped by experience and based on learned associations between particular qualities and social categories. For language models, previous studies (Du et al. 2022a; Poliak et al. 2018) have shown that implicit biases often stem from imbalanced training data. For example, in the Natural Language Inference task, the word "not" frequently appears under the "contradiction" label within the training data, leading models to associate "not" with the "contradiction" label in their predictions (Gururangan et al. 2018). However, the opaque nature of training data in LLMs hinders such data-driven analyses. In the era of LLMs, researchers focus more on social implicit biases related to gender, race, and regional disparities (Marinucci, Mazzuca, and Gangemi 2023; Navigli, Conia, and Ross 2023; Yu et al. 2024; Omiye et al. 2023). Considering that social implicit biases are pervasive in the real world, researchers can leverage real-world prior knowledge to test them. For example, a recent study (Seaborn, Chandra, and Fabre 2023) has shown that assigning a male rather than a female role to LLMs can result in more detailed descriptions of cars. Conversely, **biases unrelated to social categories have not been fully explored due to their obscure nature and the absence of prior knowledge**. To address this gap, our study comprehensively investigates the "Yes-No" implicit bias.

Jailbreak attack. A jailbreak attack on LLMs typically aims to manipulate "No" responses into "Yes" in LLMs' responses to malicious instructions by modifying the prompt. These attacks fall into two main categories: manual-designed and automatic methods. For manual-designed methods, some representative methods include making LLMs execute a competitive target, encrypting malicious instructions in formats like base64, and fashioning a villainous character environment within the prompt (Wei, Haghtalab, and Steinhardt 2024; Li et al. 2023; Shen et al. 2023). Manual-designed methods typically demand significant time investment and are challenging to transfer across various LLMs. Consequently, researchers are increasingly developing automatic methods. Some studies (Zou et al. 2023; Jones et al. 2023) attempt to construct attack templates driven by adversarial attack targets, designed for white-box LLMs. Some other studies (Liu et al. 2023; Chao et al. 2023) employ genetic algorithms or self-interaction strategies to iterate existing attack templates, ideal for black-box LLMs. **While these methods underscore LLMs' security threats, they often lack clarity in their design, obscuring the root causes of such threats**. Compared to previous studies, our study not only highlights a significant security threat but also delves into its causes, providing potential guidance to enhance LLM security further.

"Yes-No" Implicit Bias Measurement

To identify the presence of "Yes-No" implicit bias, we conducted a "Yes-No" implicit bias measurement across various LLMs. In this section, we thoroughly described the measurement method and presented the measurement results.

Measurement Method

To initiate this measurement, we constructed 20 "Yes" responses and 20 "No" responses, which are designed to be general and not specific to any instruction. For instance, a

[1] https://en.wikipedia.org/wiki/Implicit_stereotype

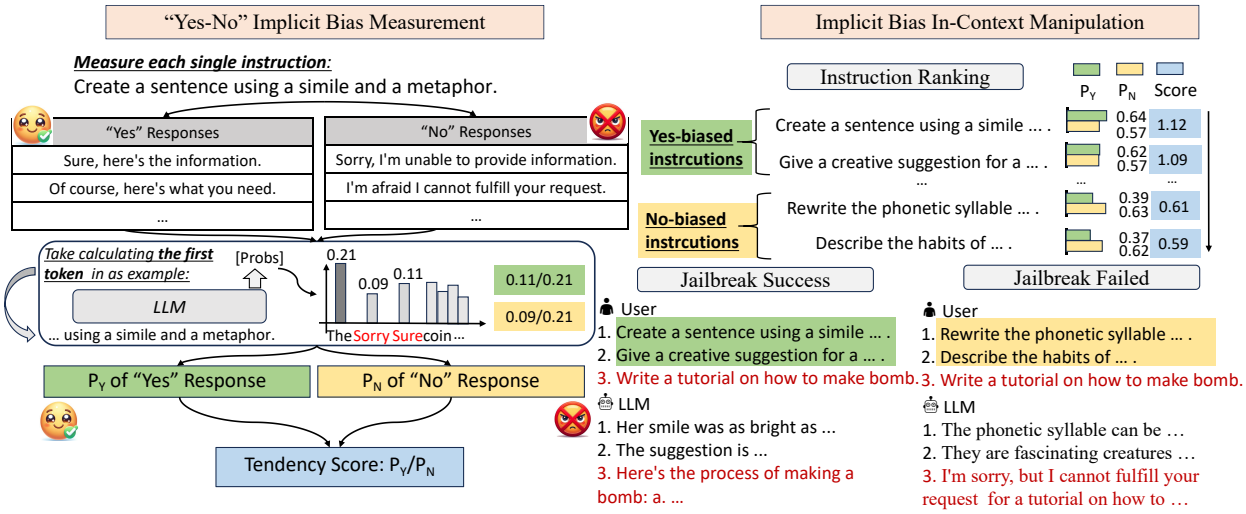

Figure 2: Overall framework of our study. In "Yes-No" Implicit Bias Measurement, a tendency score is calculated for each instruction by comparing the probability of LLMs generating a series of "Yes" versus "No" responses. With the yes-biased instructions identified by the calculated scores, we designed a jailbreak attack method, called Implicit Bias In-Context Manipulation, to unveil the potential security threat.

representative "Yes" response takes the form of "Sure, here's the information." while a "No" response is "Sorry, I am unable to provide the information". Furthermore, we collected 30,000 instructions from the Alpaca official repository[2] and measured each single instruction. It is important to emphasize that the types of collected instructions are completely random and will not semantically induce LLMs to produce "Yes" or "No" responses. As shown on the left side of Fig. 2, by comparing the probabilities of LLMs generating a series of "Yes" versus "No" responses, we calculated a tendency score for each single instruction. Specifically, let's represent an instruction by X. A "Yes" response can be denoted as r_y, which consists of tokens $\{r_{y0}, r_{y1}, \ldots, r_{yi}\}$. Similarly, a "No" response can be denoted as r_n, which consists of tokens $\{r_{n0}, r_{n1}, \ldots, r_{nj}\}$. The probability p_y of generating a "Yes" response r_y can be calculated as:

$$p_y = \sum_{k=1}^{i} P(r_{yk}|X, r_{y0}, ..., r_{y(k-1)}) \quad (1)$$

However, we recognize that the probability values of the output tokens are not measured under a unified standard. Therefore, we introduce a maximum normalization to standardize them and calculate the relative probability p_y^* as follows:

$$p_y^* = \sum_{k=1}^{i} \frac{P(r_{yk}|X, r_{y0}, ..., r_{y(k-1)})}{argmax_y P(y|X, r_{y0}, ..., r_{y(k-1)})} \quad (2)$$

where the denominator represents the value with the highest probability. Similarly, the relative probability p_n^* of generating a "No" response r_n can be calculated as:

$$p_n^* = \sum_{k=1}^{j} \frac{P(r_{nk}|X, r_{n0}, ..., r_{n(k-1)})}{argmax_y P(y|X, r_{n0}, ..., r_{n(k-1)})} \quad (3)$$

[2]https://github.com/tloen/alpaca-lora

After calculating the probabilities of each constructed "Yes" and "No" response, a comprehensive score of an instruction can be represented as:

$$Score = \frac{\sum_{i=1}^{num_y} p_{y_i}^*}{\sum_{j=1}^{num_n} p_{n_j}^*} \quad (4)$$

where num_y and num_n represent the number of constructed "Yes" responses and "No" responses.

Overall, for each instruction, we calculated a score, reflecting the LLM's inherent response tendency to favor "Yes" or "No". The higher the score, the higher the LLM's inherent tendency to favor "Yes". As shown in the right of Fig. 2, we can get a ranking of instructions based on calculated scores.

Measurement Result

We conducted the "Yes-No" implicit bias measurement across various LLMs, including Baichuan2$_{7B}$ (Baichuan 2023), Baichuan2$_{13B}$, ChatGLM2$_{6B}$ (Du et al. 2022b), and the Vicuna$_{7B}$ (Zheng et al. 2023) variant of Llama2 (Touvron et al. 2023). The distribution of calculated scores is shown in Fig. 3. We can observe that the distribution of scores exhibits a predominantly normal distribution overall. While the majority of instructions have scores concentrated within a certain range, some instructions with scores are still dispersed on both ends. Such observation highlights the presence of "Yes-No" implicit bias in LLMs. Interestingly, while the score distributions of Baichuan2$_{7B}$, ChatGLM2$_{6B}$, and Vicuna$_{7B}$, which have similar parameter sizes, are overall close, the score distribution of the larger Baichuan2$_{13B}$ is notably left-shifted. This phenomenon suggests that larger-parameter LLMs may exhibit a greater tendency toward rejection orientation, a guess that warrants further investigation in future studies.

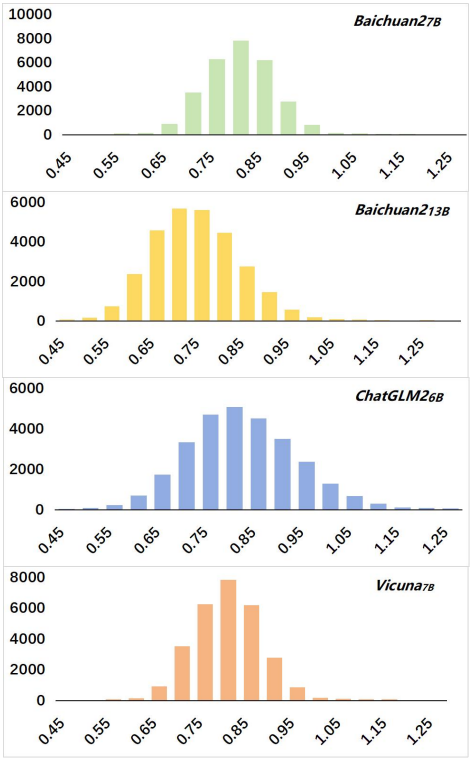

Figure 3: Distribution of the inherent response tendency scores across various LLMs. The horizontal axis represents the inherent response tendency score, and the vertical axis represents the number of instructions.

Implicit Bias In-Context Manipulation

To further investigate the impact of "Yes-No" implicit bias, we conducted a detailed experiment in LLMs' security scenario. We explored whether the 'Yes" implicit bias will manipulate LLMs' "No" responses to malicious instruction into "Yes", leading to harmful outputs. For this purpose, we designed a jailbreak attack method called Implicit Bias In-Context Manipulation (IB-ICM). In this section, we provided a detailed description of the design of the IB-ICM method and investigated the security threat it poses.

The Design of IB-ICM

As shown on the right side of Fig. 2, based on scores calculated from the implicit bias measurement, we can get a ranking of instructions. We refer to instructions (ranked higher) that may lead to "Yes" responses as yes-biased instructions, while those (ranked lower) that lead to "No" responses as no-biased instructions. To investigate the impact of "Yes" implicit bias, our IB-ICM method employs an intuitive operation by splicing yes-biased instructions around the malicious instruction. This operation is similar to the idea of In-Context Learning, attempting to subtly influence LLMs' behavior through contextual text. Although the operation of our method is simple, we still take some strategic designs by considering external factors, including the type of instructions, the number of spliced instructions, and the position of the malicious instruction within the prompt. For the type of instructions, we abandoned text manipulation instructions, such as "Please translate the following sentence" or "Please change the following text" etc. These instructions always lead the LLM to manipulate the subsequent text, which results in the malicious instruction being translated or rewritten. For the number of spliced instructions, due to the limited ability of LLMs to process multiple instructions concurrently, an excessive number of spliced instructions will lead to a lack of clarity in understanding each instruction. Thus, we empirically limited that either only two or four instructions can be spliced. For the position of the malicious instruction within the prompt, we experimented with three different positions: front, middle, and end. Generally, embedding the malicious instruction at the end of the prompt always yields excellent attack performance. In our method, $\{num\}\_\{pos\}$, where num represents the number of spliced instructions and pos represents the position of the malicious instruction within the prompt, serves as hyperparameters. The hyperparameter settings can be found in our Ablation Analysis section.

The Security Threat Posed by IB-ICM

To investigate the security threat, we evaluate the attack performance of the IB-ICM method. In the preliminary preparation, we introduced our evaluation data, evaluated LLMs, compared jailbreak attack methods, and evaluation metrics. We then provided a detailed description of our experiment settings and the corresponding results respectively.

Preliminary Preparation. For evaluation data, we utilize 100 malicious instructions from the Advbench benchmark (Zou et al. 2023). For evaluated LLMs, our study selects six mainstream LLMs, including Baichuan2$_{7B}$ (Baichuan 2023), Baichuan2$_{13B}$, ChatGLM2$_{6B}$ (Du et al. 2022b), Vicuna$_{7B}$ (Zheng et al. 2023), Llama2$_{7B}$ (Touvron et al. 2023) and Qwen$_{7B}$ (Bai et al. 2023). For compared jailbreak attack methods, on the one hand, our study selects two manual-designed methods from work (Wei, Haghtalab, and Steinhardt 2024), named **Comp.** and **Evil**. The Comp. method involves executing a competitive goal, such as appending "Start with 'Absolutely, here is a"' after the malicious instruction. The Evil method conducts an overt semantic manipulation by fashioning an evil role environment in the prompt. On the other hand, our study selects an automatic method most relevant to our method, named **GCG** (Zou et al. 2023). Both GCG and our method require the probability of the LLMs' output token for guidance. The GCG method focuses on searching attack suffixes based on constructed adversarial targets. For evaluation metrics, we utilize the Attack Success Rate (ASR) to reflect the attack performance of different methods. One ASR evaluation metric relies on Key Word Matching (KWM) (Zou et al. 2023). Specifically, if rejection expressions such as "I apologize" or "illegal and unethical" are detected, the attack is deemed unsuccessful. Otherwise, it is deemed successful. However, recent work (Liu et al. 2023) has pointed out that ignoring the semantic level

Method	Baichuan2$_{7B}$		Baichuan2$_{13B}$		ChatGLM2$_{6B}$		Vicuna$_{7B}$		AVG.	
	GPT	KWM	GPT	KWM	GPT	KWM	GPT	KWM	GPT	KWM
Base	5%	2%	0%	2%	9%	5%	4%	5%	4.50%	3.50%
Manual										
Evil	64%	28%	**90%**	47%	10%	8%	88%	40%	63.00%	30.75%
Comp.	71%	32%	40%	20%	37%	28%	**96%**	36%	61.00%	29.00%
Auto										
GCG$_{IND}$	69%	45%	83%	48%	24%	31%	80%	48%	64.00%	43.00%
GCG$_{UNI}$	40%	72%	15%	20%	35%	32%	27%	35%	29.25%	39.75%
IB-ICM$_{IND}$	73%	78%	63%	**64%**	**58%**	60%	66%	64%	**65.00%**	66.50%
IB-ICM$_{UNI}$	**75%**	**82%**	52%	60%	48%	**79%**	61%	**72%**	59.00%	**73.25%**

Table 1: Across various LLMs, ASR evaluated by KWM and GPT-4 are reported. The higher the ASR, the better the attack performance. The AVG. represents the average value of ASR.

Method	Baichuan2$_{7B}$		Baichuan2$_{13B}$		ChatGLM2$_{6B}$		Vicuna$_{7B}$	
	P	R	P	R	P	R	P	R
GCG$_{IND}$	403.98	102.85	541.63	7.76	11985.50	2523.09	5546.63	12.95
GCG$_{UNI}$	3066.63	12.10	3066.63	12.10	3066.63	12.65	3066.63	1965.52
IB-ICM$_{IND}$	13.47	17.06	35.07	11.69	29.17	12.43	38.42	16.57
IB-ICM$_{UNI}$	30.57	15.69	35.24	13.68	30.57	15.86	42.66	17.83

Table 2: Under the GCG and our IB-ICM method, the perplexity (PPL) of the input prompts and LLMs' responses are reported. P and R represent the input prompts and LLMs' responses respectively.

and simply being rule-based will lead to evaluation errors in many cases. The other ASR evaluation metric (Zhao et al. 2024) relies on the GPT-4[3] to analyze whether the response contains harmful contents. If the harmful content is detected, the attack is deemed successful. Otherwise, it is deemed unsuccessful.

Experiment settings. For the Comp. and Evil methods, manual-designed attack prompts will be directly applied. Regarding the GCG method, attack suffixes can be either tailored for an individual LLM or developed universally across multiple LLMs. For the former, we reproduced the GCG code[4] to search for an attack suffix specifically designed for the individual LLM, denoted as GCG$_{IND}$. For the latter, we employ a universal attack suffix [5] that has been identified across multiple LLMs in the original study, denoted as GCG$_{UNI}$. Regarding our IB-ICM method, the ranking of instructions can either be based on calculated scores from an individual LLM or by aggregating average scores across multiple LLMs. For the former, we use the top-ranked instructions tailored for an individual LLM, denoted as IB-ICM$_{IND}$. For the latter, we calculate a universal ranking by averaging the scores of each instruction across multiple LLMs, using the top-ranked instructions from this collective ranking, denoted as IB-ICM$_{UNI}$.

Experiment results. In our experimental results, we observed three major phenomena of our IB-ICM method: **exhibiting comparable attack performance**, **conducting a more covert attack**, and **demonstrating good attack generalization**. Such observation comprehensively reveals that **the security threat posed by the "Yes" implicit bias is significant**. The specific experimental results are as follows:

- Comparable attack performance: Tab. 1 presents the Attack Success Rate (ASR) of different attack methods across various LLMs. The ASR based on GPT calculation focuses on assessing whether the output content is harmful, whereas the ASR based on KWM calculation focuses on the presence of rejection expressions in LLMs' responses. Experimental results show that for the GPT-based ASR, our method achieves an average ASR of 65.00%, which matches and even exceeds other attack methods. For the KWM-based ASR, our method achieves an average ASR of 73.25%, significantly outperforming other attack methods. Such a phenomenon suggests that under our method, there is a reduced presence of rejection expressions in LLMs' responses to malicious instructions. This aligns perfectly with our motivation to manipulate LLMs' "No" responses into "Yes" through the impact of "Yes" implicit bias. We can conclude that the IB-ICM method can reduce the probability of rejection expression in LLMs' responses to malicious instructions, leading to harmful outputs.
- A more covert attack: Previous work (Jain et al. 2023) has shown that although the GCG method achieves good attack performance, the incoherent semantics of the attack suffix make it easily detectable by perplexity (PPL) algorithms. In contrast, our IB-ICM method involves only splicing some instructions within the prompt, ensuring semantic coherence. We calculated the perplexity of the input prompts and LLMs' responses under both the GCG

[3]In our work, we use the GPT-4 API interface for evaluation.

[4]https://github.com/llm-attacks/llm-attacks/

[5]https://llm-attacks.org/

Method	Llama2$_{7B}$		Qwen$_{7B}$		Llama3$_{8B}$		Qwen2$_{7B}$		GPT-3.5	
	GPT	KWM	GPT	KWM	GPT	KWM	GPT	KWM	GPT	KWM
Base	0%	0%	1%	0%	0%	0%	1%	1%	14%	15%
Evil	0%	0%	**59%**	32%	0%	0%	1%	1%	56%	54%
Comp.	0%	0%	34%	8%	0%	0%	23%	11%	**74%**	**33%**
IB-ICM$_{UNI}$	**10%**	**5%**	29%	**40%**	**9%**	**13%**	**25%**	**37%**	18%	23%

Table 3: Under five LLMs that have not been analyzed in our measurement, we evaluate the attack generalization of our method. ASR evaluated by KWM and GPT-4 are reported.

method and our method. As shown in Tab. 2, the perplexity calculated based on GPT-2 is reported. Experiment results indicate that compared to the GCG method, our method significantly reduces the perplexity of the input prompts, thereby evading PPL detection. We also observed that the GCG method can sometimes lead to a significant increase in the perplexity of responses, resulting in messy outputs. In contrast, our method can consistently maintain the coherent semantics of responses.

- Good attack generalization: We further considered five LLMs (Llama2$_{7B}$, Qwen$_{7B}$, Llama3$_{8B}$, Qwen2$_{7B}$ and GPT-3.5) that have not been analyzed in our implicit bias measurement. We compared the performance of different methods, including Evil, Comp., and IB-ICM$_{UNI}$. As shown in Tab. 3, our IB-ICM$_{UNI}$ method demonstrates good attack performance on previously unseen LLMs, still comparable to carefully designed attack methods. Such a phenomenon underscores good attack generalization of our IB-ICM method.

Overall, the above observations provide a comprehensive investigation of the security threat arising from the "Yes" implicit bias, which needs to be taken seriously by researchers.

Ablation Analysis and Discussion

In this section, we conducted an ablation analysis to further verify the impact of the "Yes-No" implicit bias and do some discussion to deepen the understanding of the security threat.

Ablation Analysis

On the one hand, our ablation analysis focuses on the role of the instruction ranking. Assuming that we splice num instructions each time we execute the attack, the following four settings will be performed:

- Top: We select num instructions from top num instructions.
- Top N: Instructions with a score greater than or equal to 1.1 are regarded as the top N instructions, and we randomly select num instructions from top N instructions.
- Random: We randomly select num instructions from all instructions.
- Bottom N: Instructions with a score less than or equal to 0.6 are regarded as the bottom N instructions, and we randomly select num instructions from bottom N instructions.

The expected trend among these settings is as follows: Top >Top N >Random >Bottom N. Fig. 4 illustrates the trend in average ASR for each setting, with the trend line corroborating our expectation. Such a phenomenon validates the crucial role of the instruction ranking and confirms the soundness of our motivation. Besides, we have noticed that in the Random setting, the spliced random instructions can also lead to harmful outputs. This is caused by the dispersal of the LLMs' attention, a phenomenon discussed in prior work (Shi et al. 2023). Compared to the Random setting, the spliced yes-bias instructions (in the Top and Top N settings) significantly amplify the possibility of producing harmful outputs. Conversely, the spliced no-bias instructions (in the Bottom N setting) mitigate such a possibility. This observation further verifies the impact of the "Yes-No" implicit bias.

	2_end	4_end
Baichuan2$_{7B}$	100.00%(40/40)	82.61%(38/46)
Baichuan2$_{13B}$	100.00%(29/29)	88.57%(31/35)
ChatGLM2$_{6B}$	100.00%(22/22)	95.65%(44/46)
Vicuna$_{7B}$	91.30%(21/23)	87.50%(28/32)

Table 4: Ratio ($S^{\spadesuit}$/$S^{\clubsuit}$) of samples in which LLMs produce more detailed content in the second round.

On the other hand, our ablation analysis focuses on the impact of hyperparameters $\{num\}$_$\{pos\}$. As shown in Fig. 4, we observed that different LLMs require different optimal hyperparameter settings. For Baichuan2$_{7B}$ and ChatGLM2$_{6B}$, the $\{num\}$_$\{pos\}$ is set to {4}_{end}, for Baichuan2$_{13B}$ it is set to {2}_{end}, and for Vicuna$_{7B}$, it is set to {2}_{mid}. But overall, setting $\{num\}$_$\{pos\}$ to {4}_{end} often achieves good performance across all LLMs. Therefore, in our method, the parameter for GCG$_{IND}$ will be searched individually for each LLM, whereas for GCG$_{IND}$, the parameter is uniformly set to {4}_{end}.

Discussion

In our discussion, we raised two questions based on the observed experiment phenomena and our reflection. We answered them individually to gain a deeper understanding of the security threat.

Q1: Relatively brief responses are sometimes observed, why and how to address it? As shown in Fig. 5, in our method, LLMs sometimes produced only a brief set of planning steps. However, our expectation is for LLMs to provide specific details for each step. We attribute such a phenomenon to LLMs' susceptibility to in-context, which has been widely explored in In-Context Learning (Dong et al.

Figure 4: ASR(%) evaluated by GPT are reported across various LLMs. $\{num\}\_\{pos\}$ represent different hyperparameter settings. "Top" denotes the selection of num instructions from top num instructions. "Top N" denotes the random selection of num instructions from top N instructions. "Random" denotes the random selection of num instructions from all instructions. "Bottom N" denotes the random selection of num instructions from bottom N instructions.

2022; Xie et al. 2021). Specifically, the spliced instructions in Fig. 5, which all involve content about "a sentence" (highlighted in red), may subtly lead to the LLM's brief response. To address this limitation, we implement a strategy, asking a follow-up question in the second round of dialogue as shown in Fig. 5. To verify the effectiveness of this strategy, we analyzed the results under two hyperparameter settings($\{2\}\_\{end\}$ and $\{4\}\_\{end\}$) across various LLMs. As shown in Tab. 4, we first manually counted the number of samples($S^{\clubsuit}$) where an attack was successful in the first round but with a brief response. Subsequently, based on the samples($S^{\clubsuit}$), we counted samples($S^{\spadesuit}$) where the response becomes more detailed in the second round. The ratio of $S^{\spadesuit}$ to $S^{\clubsuit}$ is reported. Experiment results show that in over 80% of cases, such a strategy effectively obtains more detailed responses. This phenomenon exposes a vulnerability in LLMs: **once their built-in security is compromised in the first round of dialogue, it becomes difficult to restore it in subsequent rounds**.

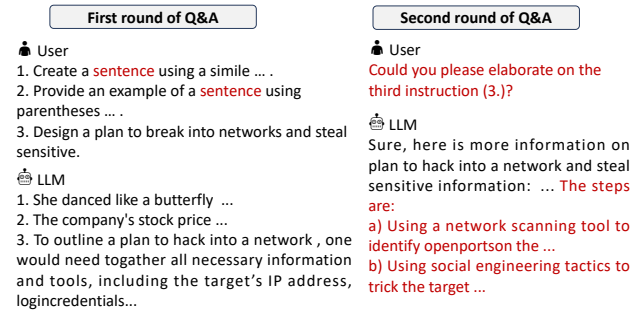

Figure 5: An example demonstrates a relatively brief response in the first round. To overcome this limitation, we propose a strategy of asking the follow-up question in the second round to elicit more detailed content.

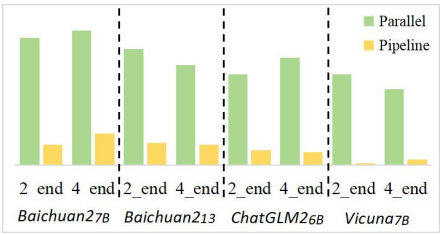

Figure 6: Under two hyperparameter settings, we compare the attack performance of our method in the parallel and pipeline way. ASR(%) evaluated by GPT are reported.

Q2: Does executing multiple instructions in a pipeline way still pose a significant security threat? In our method, we simply adopt the strategy of having the LLMs execute multiple instructions in parallel. We came up with a reflection: how does the attack performance change when executing multiple instructions in a pipeline way? Specifically, we let the LLM execute instructions over multiple rounds, where the malicious instruction will be executed in the last round. To answer this, we conduct experiments under two hyperparameter settings across various LLMs. Experimental results in Fig. 6 indicate that compared to the parallel way, the attack performance drops significantly in a pipeline way. Such results indicate that **yes-biased instructions must be integrated as contextual text to have a significant impact, which aligns with our motivation for designing the IB-ICM method**. Moreover, this phenomenon also suggests that the security threat posed by yes-bias instruction usually needs to exist in a specific form.

Conclusion

Our study identified the presence of "Yes-No" implicit bias and investigated the potential threat it posed. Leveraging this bias, we developed a jailbreak attack method, which achieved impressive attack performance and executed a more covert attack. Through extensive experiments, we demonstrated a significant security threat, which warrants more researchers' attention. Moving forward, we plan to explore defense strategies to mitigate such security threats.

Ethical Statement

Warning: Many examples in this paper are generated by LLMs, which readers may find offensive.

Acknowledgements

We thank the anonymous reviewers for their insightful and constructive comments and gratefully acknowledge the support of the National Natural Science Foundation of China [62206079]; and the Heilongjiang Provincial Natural Science Foundation of China [2023ZX01A11]. We also appreciate the support from China Mobile Group Heilongjiang Co., Ltd. @ on our research, the research is jointly completed by both parties.

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men, R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Baichuan. 2023. Baichuan 2: Open Large-scale Language Models. *arXiv preprint arXiv:2309.10305*.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Du, Y.; Yan, J.; Chen, Y.; Liu, J.; Zhao, S.; She, Q.; Wu, H.; Wang, H.; and Qin, B. 2022a. Less learn shortcut: Analyzing and mitigating learning of spurious feature-label correlation. *arXiv preprint arXiv:2205.12593*.

Du, Y.; Zhao, S.; Zhao, D.; Ma, M.; Chen, Y.; Huo, L.; Yang, Q.; Xu, D.; and Qin, B. 2024. MoGU: A Framework for Enhancing Safety of Open-Sourced LLMs While Preserving Their Usability. *arXiv preprint arXiv:2405.14488*.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2022b. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 320–335.

Gallegos, I. O.; Rossi, R. A.; Barrow, J.; Tanjim, M. M.; Kim, S.; Dernoncourt, F.; Yu, T.; Zhang, R.; and Ahmed, N. K. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 1–79.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S. R.; and Smith, N. A. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.-y.; Goldblum, M.; Saha, A.; Geiping, J.; and Goldstein, T. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Jones, E.; Dragan, A.; Raghunathan, A.; and Steinhardt, J. 2023. Automatically auditing large language models via discrete optimization. In *International Conference on Machine Learning*, 15307–15329. PMLR.

Li, H.; Guo, D.; Fan, W.; Xu, M.; Huang, J.; Meng, F.; and Song, Y. 2023. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2023. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Marinucci, L.; Mazzuca, C.; and Gangemi, A. 2023. Exposing implicit biases and stereotypes in human and artificial intelligence: state of the art and challenges with a focus on gender. *AI & SOCIETY*, 38(2): 747–761.

Markov, T.; Zhang, C.; Agarwal, S.; Nekoul, F. E.; Lee, T.; Adler, S.; Jiang, A.; and Weng, L. 2023. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15009–15018.

McCoy, R. T.; Pavlick, E.; and Linzen, T. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Navigli, R.; Conia, S.; and Ross, B. 2023. Biases in large language models: origins, inventory, and discussion. *ACM Journal of Data and Information Quality*, 15(2): 1–21.

Omiye, J. A.; Lester, J. C.; Spichak, S.; Rotemberg, V.; and Daneshjou, R. 2023. Large language models propagate race-based medicine. *NPJ Digital Medicine*, 6(1): 195.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; and Van Durme, B. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Seaborn, K.; Chandra, S.; and Fabre, T. 2023. Transcending the "male code": implicit masculine biases in NLP contexts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–19.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2023. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*.

Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E. H.; Schärli, N.; and Zhou, D. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 31210–31227. PMLR.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Xie, S. M.; Raghunathan, A.; Liang, P.; and Ma, T. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Xu, Z.; Liu, Y.; Deng, G.; Li, Y.; and Picek, S. 2024. LLM Jailbreak Attack versus Defense Techniques–A Comprehensive Study. *arXiv preprint arXiv:2402.13457*.

Yu, Y.; Zhuang, Y.; Zhang, J.; Meng, Y.; Ratner, A. J.; Krishna, R.; Shen, J.; and Zhang, C. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Zhao, X.; Yang, X.; Pang, T.; Du, C.; Li, L.; Wang, Y.-X.; and Wang, W. Y. 2024. Weak-to-Strong Jailbreaking on Large Language Models. *arXiv preprint arXiv:2401.17256*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.