Strengthened Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors

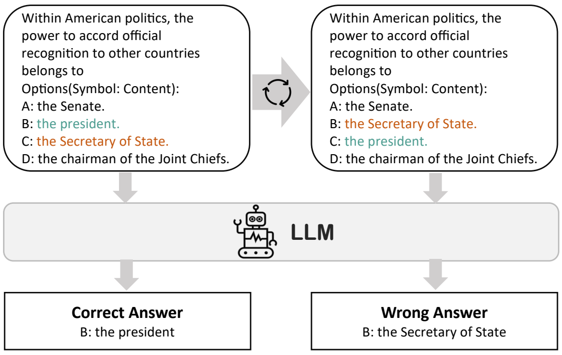
Mengge Xue[*], Zhenyu Hu[*], Liqun Liu[†],
Kuo Liao, Shuang Li, Honglin Han, Meng Zhao, Chengguo Yin
Tencent
{berryxue, mapleshu, liqunliu}@tencent.com

Abstract

Multiple-Choice Questions (MCQs) constitute a critical area of research in the study of Large Language Models (LLMs). Previous works have investigated the selection bias problem in MCQs within few-shot scenarios, in which the LLM's performance may be influenced by the presentation of answer choices, leaving the selection bias during Supervised Fine-Tuning (SFT) unexplored. In this paper, we reveal that selection bias persists in the SFT phase , primarily due to the LLM's inadequate Multiple Choice Symbol Binding (MCSB) ability. This limitation implies that the model struggles to associate the answer options with their corresponding symbols (e.g., A/B/C/D) effectively. To enhance the model's MCSB capability, we first incorporate option contents into the loss function and subsequently adjust the weights of the option symbols and contents, guiding the model to understand the option content of the current symbol. Based on this, we introduce an efficient SFT algorithm for MCQs, termed Point-wise Intelligent Feedback (PIF). PIF constructs negative instances by randomly combining the incorrect option contents with all candidate symbols, and proposes a point-wise loss to provide feedback on these negative samples into LLMs. Our experimental results demonstrate that PIF significantly reduces the model's selection bias by improving its MCSB capability. Remarkably, PIF exhibits a substantial enhancement in the accuracy for MCQs[‡].

1 Introduction

Multiple-Choice Questions (MCQs) are ubiquitously employed in the realm of Large Language models (LLMs). They typical comprise a query accompanied by an array of potential options, wherein the model's assignment is to discern and select the most appropriate solution. Given the significance, numerous studies (Pezeshkpour and Hruschka, 2023; Zheng et al., 2023a; Robinson et al., 2023) have explored the challenges faced in the few-shot phase of MCQs. One of the main challenges is the selection bias (Pezeshkpour and Hruschka, 2023; Zheng et al., 2023a), which refers that LLMs are sensitive to variations in the arrangement of options within MCQs. Figure 1 demonstrates an example of selection bias.



Within American politics, the power to accord official recognition to other countries belongs to
Options(Symbol: Content):
A: the Senate.
B: the president.
C: the Secretary of State.
D: the chairman of the Joint Chiefs.

Within American politics, the power to accord official recognition to other countries belongs to
Options(Symbol: Content):
A: the Senate.
B: the Secretary of State.
C: the president.
D: the chairman of the Joint Chiefs.

Figure 1: Selection bias of MCQs. Upon transposition of the correct content from option B to C, the model persists in selecting B instead of the correct option content.

Foundation LLMs such as LLaMA2-7B can seldom acquire data-sensitive domain knowledge. For such scenarios, it is crucial to perform Supervised Fine-Tuning (SFT) on LLMs. Thus we hope LLMs to ensure accuracy and also robustly select reliable options in MCQs during SFT. Unfortunately, we find that the selection bias still exists during the SFT phase. Following the methodology of Zheng et al. (2023a), we conduct "*answer-moving attack*" experiment by always moving the golden option content to a specific symbol, and the results are displayed in Table 1. In detail, we train two models using the training dataset, and select the best-performing model on the validation set and predict results for the test set. Then, we implement the

[*] These authors contributed equally to this work.
[†] Corresponding author.
[‡] The code of this work is available at `https://github.com/berryxue/PIF`

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4331–4344
August 11-16, 2024 ©2024 Association for Computational Linguistics

dataset	**MMLU**				
Move Golden to	**Orig**	**A**	**B**	**C**	**D**
LLaMA2-7B	54.6	65.7 (+11.1)	45.6 (-9.0)	58.4 (+3.8)	47.8 (-6.8)
LLaMA2-13B	59.2	54.8 (-4.4)	64.6 (+5.4)	56.3 (-2.9)	61.5 (+2.3)

Table 1: The accuracy results after answer-moving attack on the LLMs during the SFT process with MMLU benchmarks. Relocating the correct option content of MCQs to a specific symbol can lead to significant performance fluctuations for LLMs.

answer-moving attack by moving all the correct options to the same symbol for the test dataset and predict results again, which cause significant changes in the models' performance. For example, when moving all correct options to symbol A, the accuracy of the LLaMA2-7B model increases by 11.1, while LLaMA2-13B model decreases 4.4.

Why do LLMs show selection bias during the SFT stage? We propose a hypothesis that ***"Strengthened Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors"***. We utilize **M**ultiple **C**hoice **S**ymbol **B**inding (MCSB) capability (Robinson et al., 2023) to represent the LLM's ability to bind option symbols and their corresponding option contents, and employ **P**roportion of **P**lurality **A**greement (PPA) metric to compare the relative MCSB ability of two LLMs. Through comprehensive experimental validation, we discover that improving the LLMs' performance on the PPA metric alleviates the LLMs' performance on selection bias.

Based on the relationship between the LLMs' selection bias and its MCSB capabilities, we expect to mitigate selection bias by enhancing the model's MCSB capability. We first incorporate option contents into the loss function, guiding the model to understand the content of the current symbol. However, the results are less than satisfactory. Considering that label words are anchors (Wang et al., 2023a), we adjust the weights of the option symbols and contents in the optimization objective, termed Reweighting Symbol-Content Binding (RSCB). Subsequently, inspired by Reinforcement Learning from Human Feedback (Stiennon et al., 2020), we propose Point-wise Intelligent Feedback (PIF), in which we construct negative samples by randomly combining the contents of incorrect options with all option symbols, and design a point-wise loss to feedback these negative samples into SFT. Finally, PIF not only ensures the stability of model performance but also enhances it.

In summary, our contributions are as follows: (1) We conduct experiments to demonstrate that there is still the selection bias when performing SFT on LLMs for MCQs. We hypothesize that *Strengthened Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors*. In Section 2.3, we validate this hypothesis by investigating the correlation between the MCSB capability and the selection bias. (2) We mitigate selection bias by enhancing the model's MCSB capability. We propose the Point-wise Intelligent Feedback (PIF) method, which constructs negative samples by randomly combining the content of incorrect options with all candidate symbols and designing a point-wise loss to provide feedback on these negative samples into LLMs. (3) We conduct extensive experiments to validate that our PIF method can significantly alleviate the selection bias of LLMs during the SFT phase for MCQs. Meanwhile, the experimental results prove that PIF could also enhance the accuracy performance of the model.

2 Exploration of Selection Bias During Supervised Fine-tuning

2.1 Experimental Background

Datasets To investigate the selection bias in the SFT stage of LLMs for MCQs, we conduct experiments on several commonly used MCQs benchmarks: Massive Multitask Language Understanding(**MMLU**) benchmark (Hendrycks et al., 2021) and CommonsenseQA(**CSQA**) (Talmor et al., 2019). Our choice of benchmarks considers the variety of tasks and fields involved. Explicitly, MMLU comprises 4-option multiple-choice questions, whereas CSQA contains 5-option variants. As for domains, MMLU encompasses a wide range of tasks from fields including STEM, humanities, social science and others, coverage 57 subjects; while CSQA conduct questions draw upon ConceptNet (Speer et al., 2017) concepts, enabling variety and integration of worldly knowledge within the queries. It's worth noting that the CSQA dataset does not have standard answers for the test data. Thus in our experimental setup, we use the dev set as the test set and extract a portion of samples from the train set to serve as the validation set. MMLU is additionally split into four domains (STEM, Social Science, Humanities, Others) based on its subject categories. Details can be found in Appendix A.

Models We conduct experiments on LLMs origi-

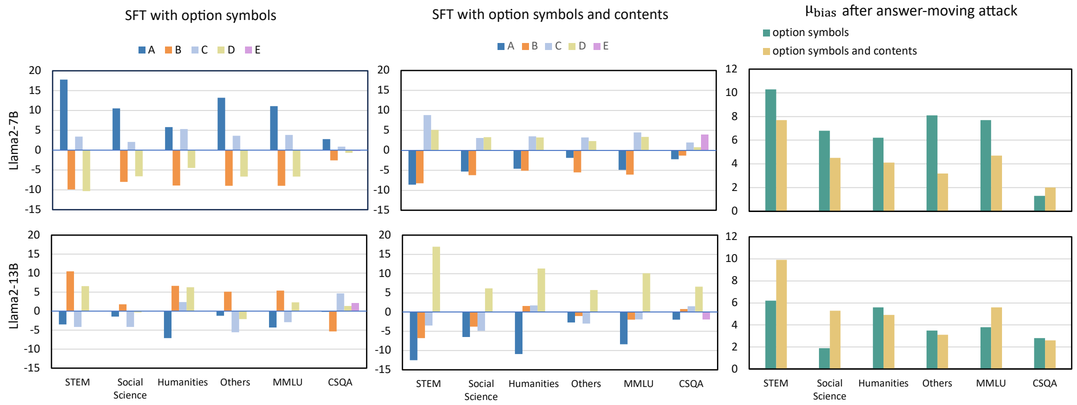


Figure 2: LLMs' selection bias during SFT. The leftmost two columns demonstrate the changes in accuracy following the answer-moving attack, whereas the rightmost column exhibits the metric μ_{bias} as defined by Equation 1.

nating from well-known LLM families, spanning various sizes. **LLaMA** (Touvron et al., 2023) constitutes a compilation of widely-used foundation models designed to promote research on LLMs for training, inference, and extensive applications. In this paper, taking into account the model size, we assess the 7B and 13B variant of LLaMA2 (LLaMA2-7B, LLaMA2-13B). Projects used in this work is illustrated in Appendix B.

Taking into account the limited computational resources, we fine-tune LLMs with **L**ow-**R**ank **A**daptation(LoRA) (Hu et al., 2022) and set its rank to 16, the alpha parameter to 64, and the dropout rate to 0.1. To investigate the selection bias, we design two distinct output configurations for LLMs during the training process. One generates only the option symbols, defined as π_{Symbol}, while the other produces both the option symbols and contents concurrently, defined as π_{SCB}. During the testing process, the model is configured to output only the symbol, which is sufficient for determining whether the answer is correct or not.

2.2 Selection Bias During SFT

We conduct experiments using two LLMs on six datasets. Similar to Zheng et al. (2023a), we conduct *"answer-moving attack"* experiment to measure the selection bias. During testing, we move all the correct answers to A|B|C|D|E respectively (Answers are A|B|C|D in MMLU. For the sake of convenience, we will use the unified notation of A|B|C|D|E) for the test set, and then display the model's accuracy after the relocation. Subsequently, to provide a concrete numerical representation of the impact of the answer-moving attack, we calculate μ_{bias}, the average absolute value of the difference between the accuracy after the answer-moving attack and the original accuracy tested on the standard test set. Which can be formulated as:

$$\mu_{\text{bias}} = \frac{\sum_{i=1}^{K}(|\text{Acc}_i - \text{Acc}_0|)}{K}, \qquad (1)$$

Acc_i refers to the accuracy after answer-moving attack, Acc_0 is the accuracy on the standard testing set, K is the number of options, and is set as 5 for CSQA, 4 for MMLU. Through the experimental results in Figure 2, we make following observations: **Selection bias varies among different sizes of the model parameters.** As seen from Figure 2, LLaMA2-7B tends to choose C, while LLaMA2-13B prefers to choose D. Although these two models belong to the same model family, their performances are not identical, which aligns with the findings in Paper Zheng et al. (2023a). We also observe that, when fine-tuning the model only with option symbols, LLaMA2-13B has a smaller μ_{bias} on MMLU compared to LLaMA2-7B, while reversed on the CSQA dataset. Therefore, during SFT, there is no absolute relationship between selection bias and the size of the model parameters.

SFT training examples influence both the magnitude and distribution of selection bias. Regardless of the LLMs, the overall bias value μ_{bias} for CSQA is always smaller than that of MMLU. As seen from Appendix A, the proportions of the five options in the CSQA benchmark training set are roughly equal, while this is not the case for MMLU. Additionally, when we fine-tune LLaMA2-13B with option symbols, the model shows a stronger inclination to predict option B on MMLU, while on CSQA, it predicts B with a significantly lower probability than others. Therefore, we speculate

Model	STEM		Social Science		Human		Others		MMLU		CSQA	
	$\Delta\mu_{\text{bias}}$	$\Delta\mu_{\text{ppa}}$	$\Delta\mu_{\text{bias}}$	$\Delta\mu_{\text{ppa}}$	$\Delta\mu_{\text{bias}}$	$\Delta\mu_{\text{ppa}}$	$\Delta\mu_{\text{bias}}$	$\Delta\mu_{\text{ppa}}$	$\Delta\mu_{\text{bias}}$	$\Delta\mu_{\text{ppa}}$	$\Delta\mu_{\text{bias}}$	$\Delta\mu_{\text{ppa}}$
LLaMA2-7B	-2.6	+0.6	-2.3	+1.2	-2.1	+2.4	-4.9	+1.2	-3.0	+1.6	+0.7	-0.2
LLaMA2-13B	+3.7	-2.1	+3.4	-1.0	-0.7	+0.2	-0.4	-0.3	+1.8	-0.7	-0.2	+0.3

Table 2: Relationships between $\Delta\mu_{\text{bias}}$ and $\Delta\mu_{\text{ppa}}$. When $\Delta\mu_{\text{bias}}$ is positive, $\Delta\mu_{\text{ppa}}$ is negative and vice versa. Such indicating that "*Strengthened Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors*".

that the selection bias during SFT is closely related to the fine-tuning dataset used in the SFT stage.

2.3 Why Do LLMs Suffer Selection Bias in MCQs' SFT

After analyzing the selection bias during SFT in various LLMs and datasets, we now focus on understanding why LLMs exhibit selection bias during the SFT stage. We identify two reasons why LLMs may choose the wrong answer. Firstly, the LLM does not know the correct option, which can be attributed to its capability. Secondly, the LLM is aware of the correct option content. However, selection bias makes it choose a preferred option symbol instead of one corresponding to the correct option. This indicates the model's weak ability to associate option content with the appropriate symbol. According to it, we propose a hypothesis: ***Strengthened Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors.***

In this paper, we utilize ***M****ultiple* ***C****hoice* ***S****ymbol* ***B****inding* (MCSB) capability (Robinson et al., 2023) to represent the LLM's ability to bind option symbols and their corresponding contents. Similar to Robinson et al. (2023), we also employ the ***P****roportion of* ***P****lurality* ***A****greement* (μ_{ppa}) metric to compare the relative MCSB ability of two LLMs.

$$\mu_{\text{ppa}} = \frac{1}{|\mathbb{D}|}\sum_{\mathbb{D}} \frac{\max\limits_{k\sim K}(\sum\limits_{j=1}^{K!} y_j = o_k)}{K!}. \tag{2}$$

As shown in Table 15, given a question with K options, there are $K!$ distinct arrangements of these options in a fixed set of symbols. For ease of expression, we use o_k to represent the k-th option content and y_j to describe the content of the predicted arrangement. During testing, we present each question to the model using each unique ordering, and then PPA for this question is the frequency corresponding to the most frequently predicted option content. For a dataset $\mathbb{D}$, μ_{ppa} is calculated as the average of the PPAs for all individual questions.

To demonstrate the relationship between the MCSB capability and the selection bias, we calculate the differences of μ_{ppa} and μ_{bias} between π_{Symbol} and π_{SCB}, which are defined as follows:

$$\Delta\mu_{\text{bias}} = \mu_{\text{bias}}^{\pi_{\text{SCB}}} - \mu_{\text{bias}}^{\pi_{\text{Symbol}}}, \tag{3}$$

$\Delta\mu_{\text{ppa}}$ is defined in a similar approach. As delineated in the prior analysis, if there exists a correlation between selection bias and the LLM's MCSB capacity, then a positive $\Delta\mu_{\text{ppa}}$ corresponds to a negative $\Delta\mu_{\text{bias}}$, indicating that increasing the MCSB capability resulting a reduction in the LLM's bias. We conduct experiments with LLaMA2-7B and LLaMA2-13B. The result is illustrated in Table 2. Except for the performance of LLaMA2-13B on MMLU-Others, all the remaining results indicate that when $\Delta\mu_{\text{bias}}$ is positive, $\Delta\mu_{\text{ppa}}$ is negative and vice versa. Additionally, the performance of LLaMA2-13B on MMLU-Others can be considered as having a relatively constant μ_{ppa}, rather than being contrary to our conclusion. Appendix H also demonstrates a theoretical proof.

3 Methodology

According to the previous analysis, we can mitigate the selection bias of LLMs during SFT by enhancing their MCSB abilities.

3.1 Symbol-Content Binding

Initially, we incorporate the option symbols as the LLM's target tokens during training, resulting in a model π_{Symbol}. Given input x and output y^{Symbol}, we define optimization objective as:

$$\mathcal{L}_{\text{Symbol}} = -\frac{\sum_t \log P_{\pi_{\text{Symbol}}}(y_t^{\text{Symbol}}|x, y_{<t}^{\text{Symbol}})}{|y^{\text{Symbol}}|}. \tag{4}$$

As shown in Table 1, π_{Symbol} suffers from severe selection bias. Since enhancing the MCSB capabilities can effectively alleviate the selection bias, given output y^{SCB}, we propose the **S**ymbol-**C**ontent **B**inding (SCB) debiasing method π_{SCB}, which incorporates both the option symbols and contents as the LLM's target tokens during training:

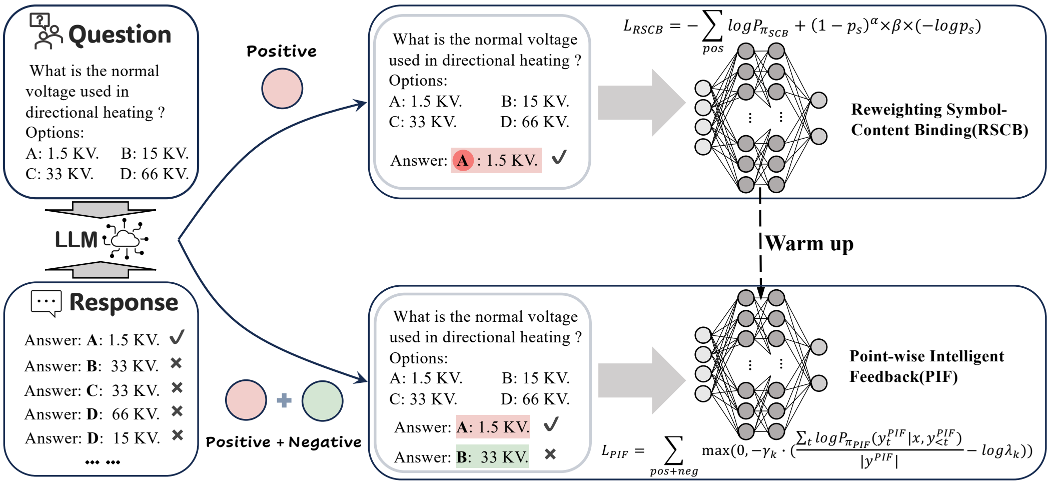

Question
What is the normal voltage used in directional heating ?
Options:
A: 1.5 KV. B: 15 KV.
C: 33 KV. D: 66 KV.
LLM
Response
Answer: A: 1.5 KV. ✓
Answer: B: 33 KV. ✕
Answer: C: 33 KV. ✕
Answer: D: 66 KV. ✕
Answer: D: 15 KV. ✕
... ...
Positive
Positive + Negative
What is the normal voltage used in directional heating ?
Options:
A: 1.5 KV. B: 15 KV.
C: 33 KV. D: 66 KV.
Answer: A : 1.5 KV. ✓
$L_{RSCB} = -\sum_{pos} logP_{\pi_{SCB}} + (1-p_s)^{\alpha}\times\beta\times(-logp_s)$
Reweighting Symbol-Content Binding(RSCB)
Warm up
What is the normal voltage used in directional heating ?
Options:
A: 1.5 KV. B: 15 KV.
C: 33 KV. D: 66 KV.
Answer: A: 1.5 KV. ✓
Answer: B: 33 KV. ✕
Point-wise Intelligent Feedback(PIF)
$L_{PIF} = \sum_{pos+neg} \max(0, -\gamma_k \cdot (\frac{\sum_t logP_{\pi_{PIF}}(y_t^{PIF}|x, y_{<t}^{PIF})}{|y^{PIF}|} - log\lambda_k))$

Figure 3: Visualization of RSCB and PIF. RSCB adjusts the weights of the option symbols and contents in the SFT optimization objective. PIF constructs negative samples by randomly combining the content of incorrect options with all option symbols and designs a point-wise loss to feedback these negative samples into SFT.

$$\mathcal{L}_{\text{SCB}} = -\frac{\sum_t \log P_{\pi_{\text{SCB}}}(y_t^{\text{SCB}}|x, y_{<t}^{\text{SCB}})}{|y^{\text{SCB}}|}. \quad (5)$$

However, the results are not as expected. As shown in Table 2, there is no clear pattern indicating that π_{SCB} has a lower bias compared to π_{Symbol}.

3.2 Reweighting Symbol-Content Binding

Actually, label words are anchors (Wang et al., 2023a), which LLMs pay more attention to. On the other hand, the answer content merely plays an auxiliary role, assisting the model in comprehending the actual content of the corresponding symbol. Thus we adjust the weights of the option symbols and contents in the optimization objective, termed **R**eweighting **S**ymbol-**C**ontent **B**inding (RSCB). The objective function is defined as:

$$\mathcal{L}_{\text{RSCB}} = \mathcal{L}_{\text{SCB}} + (1 - p_{\text{s}})^{\alpha} \cdot \beta \cdot (-\log p_{\text{s}}). \quad (6)$$

Where p_{s} is the predicted probability of the symbol token, β is the re-assigned weight for the symbol token. Considering that LLM itself has already focused on the symbol tokens for simple samples, there is no need to emphasize the symbol tokens specifically in such cases. Therefore, we ultimately employ the Focal loss (Lin et al., 2017).

3.3 Point-wise Intelligent Feedback

Human feedback allows LLMs to identify issues with accuracy, fairness, and bias (Liu et al., 2024). Previous studies have explored how to incorporate human feedback with various stages during LLM's training process, such as pre-training (Korbak et al., 2023), SFT (Yuan et al., 2023), Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al., 2020; Xue et al., 2023), and others.

In the context of MCQs, we possess knowledge of both the positive options' symbols and contents, as well as the negative options. We can also easily acquire negative symbol-content binding examples. We call this process *Intelligent Feedback* without human preference annotations. Moreover, our feedback is intrinsically point-wise, i.e., with absolute scores, the reward score for positive samples should be 1 and be λ for negative samples. We refer to this approach as **P**oint-wise **I**ntelligent **F**eedback (PIF). In this method, we aim to maximize the probability of positive examples, approaching 1, which can be optimized by cross-entropy loss, and minimize the likelihood of negative examples falling below λ. Inspired by RRHF (Yuan et al., 2023), we optimize the object of negative samples as follows:

$$\begin{aligned}\mathcal{L}_{\text{PIF}_\text{n}} = \max(0, \\ \log\lambda - \frac{\sum_t \log P_{\pi_{\text{PIF}}}(y_t^{\text{PIF}_\text{n}}|x, y_{<t}^{\text{PIF}_\text{n}})}{|y^{\text{PIF}_\text{n}}|}).\end{aligned} \quad (7)$$

Consequently, the overall optimization objective can be summarized as follows:

$$\begin{aligned}\mathcal{L}_{\text{PIF}} = \max(0, -\gamma_k \cdot \\ (\frac{\sum_t \log P_{\pi_{\text{PIF}}}(y_t^{\text{PIF}}|x, y_{<t}^{\text{PIF}})}{|y^{\text{PIF}}|} - \log\lambda_k)),\end{aligned} \quad (8)$$

Model	Method	STEM		Social Science		Humanities		Others		MMLU		CSQA	
		$\mu_{\text{bias}}\downarrow$	$\mu_{\text{ppa}}\uparrow$	$\mu_{\text{bias}}\downarrow$	$\mu_{\text{ppa}}\uparrow$	$\mu_{\text{bias}}\downarrow$	$\mu_{\text{ppa}}\uparrow$	$\mu_{\text{bias}}\downarrow$	$\mu_{\text{ppa}}\uparrow$	$\mu_{\text{bias}}\downarrow$	$\mu_{\text{ppa}}\uparrow$	$\mu_{\text{bias}}\downarrow$	$\mu_{\text{ppa}}\uparrow$
LLaMA2-7B	Symbol	10.3	70.9	6.8	81.8	6.2	81.3	8.1	80.2	7.7	78.9	1.3	93.1
	SCB	7.7	71.5	4.5	83	4.1	83.7	3.2	81.4	4.7	80.5	2.0	92.9
	RSCB	3.8	71.8	3.4	83.1	5.3	81.8	2.9	81.6	3.0	79.8	2.4	92.4
	PIF	3.7	72.3	2.3	83.9	3.9	83.1	1.3	82.9	2.7	80.8	1.3	93.0
		(-6.6)	(+1.4)	(-4.5)	(+2.1)	(-2.3)	(+1.8)	(-6.8)	(+2.7)	(-5.0)	(+1.9)	(0)	(-0.1)
LLaMA2-13B	Symbol	6.2	73.9	1.9	84.9	5.6	79.5	3.5	83.9	3.8	80.5	2.8	92.5
	SCB	9.9	71.8	5.3	83.9	4.9	79.7	3.1	83.6	5.6	79.8	2.6	92.8
	RSCB	2.8	74.7	1.3	85.9	6.8	79.1	1.6	84.8	3.1	80.9	1.9	92.6
	PIF	4.8	74.5	2.0	85.1	2.9	80.9	2.6	84.2	3.0	81.0	0.9	93.5
		(-1.4)	(+0.6)	(+0.1)	(+0.2)	(-2.7)	(+1.4)	(-0.9)	(+0.3)	(-0.8)	(+0.5)	(-1.9)	(+1.0)

Table 3: The metric μ_{bias} (A reduced value implies a diminished selection bias) and μ_{ppa} (An elevated value suggests an enhanced MSCB capability) of four methods. The implementation of PIF methodology has effectively enhanced the model's MCSB capability across virtually all datasets, consequently alleviating the selection bias.

where k is utilized to differentiate between the positive and negative samples. In our experiments, $\gamma_{\text{pos}} = 1$, $\gamma_{\text{neg}} = -1$, $\lambda_{\text{pos}} = 1$, λ_{neg} is a hyper-parameter defined in Section 4.1.

We also find that the performance of π_{PIF} is related to its initial parameters. We ultimately initialize π_{PIF} using the parameters from π_{RSCB}. Appendix D demonstrates the ablation study.

4 Experiment

4.1 Implementation Detail

Samples For PIF When employing PIF, the inclusion of negative samples in the optimization objective is essential. Given that the ultimate goal is to enhance the LLMs' MCSB performance, negative samples are constructed by randomly combining the incorrect option contents with all candidate option symbols. For instance, we can construct a negative sample "*B: 33KV.*" for the example presented in Figure 3. Due to the limited computational resources, we randomly select one negative sample for each instance. What's more, in our experiment, the parameters of π_{PIF} are warmed up from π_{RSCB}.

Metrics We finally employ four evaluation metrics. μ_{bias} is defined by Equation 1 to measure the LLM's selection bias. μ_{ppa} is defined by Equation 2 to represent the LLM's MCSB capability. "Acc" refers to the LLM's accuracy performance on the standard test set of the current benchmark. Acc_{min} is the minimum accuracy after performing the answer-moving attack, in which case LLMs with higher Acc_{min} exhibit greater robustness.

Fine-tuning Hyper-parameters and prompts used in our experiment can be found in Appendix C.

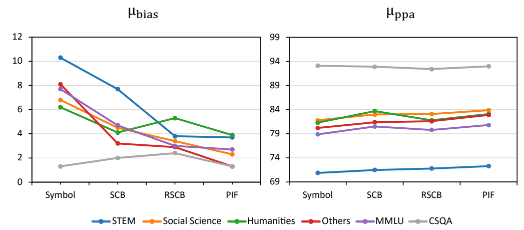


Figure 4: With the evolution of methods, μ_{bias} is gradually decreasing, μ_{ppa} is progressively increasing.

4.2 Main Results

Table 3 contains the main experimental results of comparison between four methods defined in Section 3, which are called "*Symbol*", "*Symbol-Content Binding (SCB)*", "*Reweighting Symbol-Content Binding (RSCB)*" and "*Point-wise Intelligent Feedback (PIF)*". Figure 4 demonstrates a more apparent trend in the results' changes. Appendix G also reports the accuracy after the answer-moving attack for each method on each benchmark.

The SCB method combines the option symbols and option contents as the prediction target tokens. However, this method can not reduce selection bias or enhance the LLM's MCSB capabilities across all datasets and LLMs. In fact, SCB can even exacerbate selection bias for LLaMA2-13B.

Conversely, RSCB balances the weight of option symbols and contents in the loss function based on SCB. This method enhances the abilities of MCSB on almost all datasets, which helps to reduce selection bias. This finding implies that symbols play a more significant role in MCQs learning process.

Finally, we conduct experiments on our PIF method. This method further reduces selection

Model	Method	STEM		Social Science		Humanities		Others		MMLU		CSQA	
		Acc	Acc_{min}	Acc	Acc_{min}	Acc	Acc_{min}	Acc	Acc_{min}	Acc	Acc_{min}	Acc	Acc_{min}
LLaMA2-7B	Symbol	43.8	33.5	62.5	54.5	52.2	43.3	60.6	51.6	54.6	45.6	**79.8**	77.2
	SCB	43.6	35.0	62.1	55.9	50.5	45.4	60.4	54.9	53.8	47.7	78.4	76.2
	RSCB	43.2	39.0	62.3	56.8	52.1	**45.5**	60.5	56.8	54.4	49.0	79.7	74.9
	PIF	**44.0**	**39.5**	**62.7**	**59.5**	**53.1**	42.4	**60.9**	**59.3**	**55.0**	**49.5**	79.4	**77.3**
LLaMA2-13B	Symbol	46.9	42.3	68.1	64.0	55.0	47.9	65.5	59.9	59.2	54.8	81.5	76.2
	SCB	**48.7**	36.2	68.5	62.0	54.2	43.3	64.8	61.8	58.6	50.2	81.0	79.0
	RSCB	47.5	41.8	68.0	65.2	55.8	44.3	64.6	61.8	58.7	54.3	80.9	77.9
	PIF	48.0	**42.4**	**68.5**	**65.8**	**58.4**	**52.2**	**65.6**	**62.0**	**60.0**	**56.8**	**82.2**	**79.9**

Table 4: The Acc (accuracy of standard test dataset) and Acc_{min} (Minimum accuracy after performing an answer-moving attack) of our four methods. By effectively constructing negative samples, the PIF method contributes to a notable increase in accuracy. Simultaneously, by alleviating the LLM's selection bias, it demonstrates a smaller decline in accuracy when faced with answer-moving attack, resulting in a higher Acc_{min} compared to other methods.

bias for nearly all datasets. It is noteworthy that for LLaMA2-7B's performance on CSQA and LLaMA2-13B's performance on the Social Science benchmark, the inherent bias of the models is already relatively small, at 1.3 and 1.9, respectively. Under these circumstances, the bias of our PIF method is essentially on par with the bias of the *Symbol* method. All these results validate the effectiveness of our constructed negative samples and the proposed point-wise objective function.

Moreover, based on the results presented in Table 3, it is evident that an enhancement in MCSB capability leads to a reduction in selection bias. This finding reaffirms our initial hypothesis:*Strengthened Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors*.

4.3 Impact of Our Methods on Accuracy

Essentially, while ensuring the stability of LLMs in MCQs, we also aspire to achieve higher accuracy. In Table 4, we present the performance of four methods in terms of accuracy and also the minimum accuracy after the model has been subjected to an answer-moving attack. This aims to demonstrate whether the current method can guarantee accuracy when faced with adversarial attacks.

As we can see, the simple SCB method can not lead to an improvement in accuracy, and even causes a decrease in accuracy on LLaMA2-13B. On the other hand, by emphasizing the importance of symbols, RSCB achieves a similar level of accuracy as the method trained solely on symbols. Moreover, due to the significant reduction of selection bias in the RSCB method, it outperforms *Symbol* in terms of the Acc_{min} metric.

We are pleased to report that the PIF model outperforms the other three methods in terms of both Acc and Acc_{min} metrics, yielding the most favorable results. This outstanding performance is attributed to the process of randomly selecting contents from incorrect options and combining them with all symbols while creating negative instances in the PIF model. This approach not only highlights incorrect binding relationships but also exposes the wrong option contents to the model, resulting in an enhancement of the model's accuracy.

4.4 Discussion

What would happen if there were no bias in SFT training data? Although PIF has mitigated selection bias, there is a more aggressive debiasing approach. Considering that enhancing the model's MCSB capability can alleviate selection bias, similar to the calculation of μ_{ppa}, we randomly combine symbols and contents to obtain all $K!$ possible arrangements as training data and train the model. We refer to this method as "*Perm*". Due to the limited computational resources, we randomly select $\frac{1}{K!}$ of the original training set for Perm. The results are shown in Appendix H. Perm indeed significantly alleviates selection bias, demonstrating that LLM's selection bias during SFT primarily stems from the training data utilized in the SFT. This is also consistent with the conclusion in Section 2.2.

It is practically impossible to implement the *Perm* approach as it requires using all possible random combinations, which is unrealistic due to the high computational resources and training time required. In contrast, PIF can effectively alleviate the selection bias by introducing just one negative example during the SFT process. This further confirms the effectiveness of our approach.

PIF *vs.* Data Argumentation. *Perm* method can significantly reduce selection bias, but this approach becomes unfeasible due to its high computational resources. How about we only use a portion of the data in PIF for Data Augmentation? The resources consumed by this method are comparable to those of PIF. To verify the feasibility of this method, which is called *Argum.*, we add an additional experiment based on LLaMA2-7B. Considering that PIF generates a random negative sample for each sample during training, we also augmented the original training data by randomly shuffling the combination of symbols and contents for each sample. The results are as Tabel 7. As can be seen, under the same time consumption, simple data augmentation does not perform better than PIF. Since our PIF method is derived from RLHF, the rationale for its effectiveness is the same as that of RLHF, we introduce negative feedback information to the model, informing it that such negative examples cannot be generated, thereby improving the model's performance.

Point-wise *vs.* Pair-wise. We propose Point-wise Intelligent Feedback to resume the constructed negative samples. How about incorporating these feedback data using a Pair-wise method? We implement DPO (Rafailov et al., 2023), and the results in Appendix F show that PIF outperforms both alleviating selection bias and achieving higher accuracy compared to DPO. The results prove that it is rational to make the prediction scores of the negative examples approach a minimal value λ. However, when the samples can be optimized with absolute scores, the pair-wise method will lose the information of the gap between the pairs (Cai et al., 2023).

5 Related Work

Large Language Models In recent years, the scale of the model parameters has progressively increased with the rapid development of deep learning, from 1.5 billion in 2018 (Radford et al., 2018) to 540 billion in 2022 (Chowdhery et al., 2023). A significant number of large language models have swiftly surfaced, especially after the emergence of chatGPT(OpenAI, 2022), such as LLaMA(Touvron et al., 2023), Vicuna(Chiang et al., 2023), and BLOOM(Workshop et al., 2022). These large language models have a vast amount of parameters. After being trained with massive data using generative methods, they possess the impressive ability of natural language understanding and can follow human instructions effectively (Ouyang et al., 2022).

Selection Bias in Multiple-Choice Questions The robustness and vulnerabilities of large language models have always been an important research realm. Many researchers have explored how large language models are influenced by modifications or adversarial attacks that impact individual instances in few-shot learning. For example, Zhao et al. (2021) found that large language models are easily affected by changes in task instructions and context when performing tasks. Wang et al. (2023b) and (Zheng et al., 2023b) reveal that GPT-4 tends to choose the option in the first position.

MCQs are widely used to assess the capabilities of large language models. Various MCQs datasets are introduced as standard language model benchmarks, such as MMLU(Hendrycks et al., 2021), C-Eval(Huang et al., 2023), CSQA(Talmor et al., 2019). LLMs have achieved human-like performances on various MCQ benchmarks. However, many researchers have noticed that LLMs suffer from selection bias in MCQS. Pezeshkpour and Hruschka (2023) shows that large language models are sensitive to the order of choices. Zheng et al. (2023a) found that LLMs are vulnerable to option symbol changes in MCQs due to their inherent "token bias". Robinson et al. (2023) shows that a model with high multiple choice symbol binding(MCSB) ability performs better in MCQs. However, previous studies have only explored the selection bias of LLMs in few-shot scenarios. Our work demonstrates that the selection bias still exists in SFT and mainly stems from the LLM's inadequate multiple choice symbol binding capability.

Human Feedback in LLMs OpenAI (2022) and Schulman et al. (2017) have demonstrated the potential of applying reinforcement learning in LLMs and its impressive performance. One part of the main techniques is learning a reward function from human feedback for reinforcement learning, which is often dubbed as RLHF(Christiano et al., 2017) (MacGlashan et al., 2017) (Lee et al., 2021). However, RLHF is a complex, expensive, and often unstable procedure. Lee et al. (2023) offers a promising alternative that uses a powerful LLM to generate preference instead of human annotators. To mitigate the problem of PPO's sizeable computational cost, Rafailov et al. (2023) introduces a stable, computationally lightweight method to solve the standard RLHF problem. Liu et al. (2024) convert all types of feedback into sequences of sentences, which are then used to fine-tune the model to learn human preference. The methods above mainly focus on pair-wise preference data. To use point-wise preference data, Cai et al. (2023) develop a point-wise preference learning method. In this paper, we propose PIF which designs a point-wise loss to incorporate negative samples into SFT.

6 Conclusion

This paper highlights that selection bias persists in the SFT of LLMs in MCQs. To explore why LLMs suffer from selection bias, we suppose that "Symbol Binding Makes Large Language Models Reliable Multiple-Choice Selectors". We utilize MCSB capability to represent the LLMs' ability to bind option symbols and the corresponding content and design experiments to establish the relationship between MCSB capability and the selection bias. Finally, we eliminate selection bias by enhancing the model's MCSB capability. We propose PIF, constructing negative samples by randomly combining the content of incorrect contents with all candidate symbols and designing a point-wise loss to resume these negative samples. Comprehensive experimental results demonstrate that PIF significantly reduces selection bias and substantially improves the accuracy of LLMs for MCQs.

7 Limitations

Due to limited computational resources, there are some deficiencies in our work and we list here for future reference. Firstly, we conduct experiments on LLMs with 7B and 13B parameter sizes. In the future, we are eager to understand the patterns of selection bias during the SFT phase in LLMs with even larger parameter sizes such as 70B. Secondly, during our experiments, we select one negative sample randomly for each instance. We would like to investigate the correlation between the severity of selection bias and the percentage of negative samples introduced. Finally, our method has a broader range of applications. For instance, when using LLMs as human preference annotators, they prefer the responses in a specific position. In this case, we can assign some symbols to all responses and then use PIF method to eliminate the bias. We leave these limitations for future work.

References

Tianchi Cai, Xierui Song, Jiyan Jiang, Fei Teng, Jinjie Gu, and Guannan Zhang. 2023. Ulma: Unified language model alignment with demonstration and point-wise human preference. *arXiv preprint arXiv:2312.02554*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Kimin Lee, Laura M Smith, and Pieter Abbeel. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6152–6163. PMLR.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2024. Chain of hindsight aligns language models with feedback. *International Conference on Learning Representations(ICLR)*.

James MacGlashan, Mark K. Ho, Robert Loftin, Bei Peng, Guan Wang, David L. Roberts, Matthew E. Taylor, and Michael L. Littman. 2017. Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2285–2294. PMLR.

OpenAI. 2022. https://chat.openai.com.chat, 2022.

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

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *ArXiv*, abs/2305.17926.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Wanqi Xue, Bo An, Shuicheng Yan, and Zhongwen Xu. 2023. Reinforcement learning from diverse human preferences. *arXiv preprint arXiv:2301.11774*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: Rank responses to align language models with human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023a. Large language models are not robust multiple choice selectors. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Statistics of Benchmarks

Statistics of all benchmarks used in our paper is illustrated in Table 5.

B LLMs Project

LLaMA2-7B https://huggingface.co/meta-llama/Llama-2-7b-hf
LLaMA2-13B https://huggingface.co/meta-llama/Llama-2-13b-hf
Implementation Framework
https://github.com/hiyouga/LLaMA-Factory

C Implementation Details

Fine-tuning Hyper-parameters We assign α and β of $\mathcal{L}_{\text{RSCB}}$ to 2 and 0.1, respectively. For λ of $\mathcal{L}_{\text{PIF}}$, we conduct experiments with a set of values $\{0.0001, 0.001, 0.01, 0.1\}$, and select the one that achieves the best performance on the validation set. We configure the sequence length, epoch, the maximum number of new tokens generated as 1024, 3, 4, respectively. For learning rate, we experiment with the value set $\{1e-5, 5e-5, 1e-4, 2e-4\}$, and select the one that yielded the best performance on validation. In most cases, it is set to $1e-4$. We conduct experiments on 8 GPUs. For the MMLU benchmark, for LLaMA2-7B and LLaMA2-13B models, the batch size per device is set to 4 and 2, respectively, while the gradient accumulation step is set to 2 and 4, ensuring a final total batch size of 64. For CSQA, we adjust the gradient accumulation step to achieve a final total batch size of 32. Additional details can be found in our code.

Prompts We don't use specific prompt engineering in our experiments. Instead, all the experiments are conducted using a simple prompt:
"*The following are multiple choice questions. You should directly answer the question by choosing the correct option.*
Question: {{text}}
Options: {{text}}
Answer: {{text}}"

D Ablation Study on PIF Parameter Initialization

In this part, we investigate the effectiveness of the parameter initialization for π_{PIF}, the results are contained in Table 6. Compared to PIF$_{\text{raw}}$, our proposed PIF$_{\text{RSCB}}$ performs better in terms of selection bias, MSCB capability and accuracy.

E Results of Permutation

Figure 5 illustrates the selection bias of Perm. Perm indeed significantly alleviates selection bias, which demonstrates that LLM's selection bias during SFT primarily stems from the training data for SFT.

F Results of The Point-wise and Pair-wise Method

Table 8 demonstrates the different results between point-wise method PIF and pair-wise method DPO. PIF outperforms both alleviating selection bias and achieving higher accuracy compared to DPO.

G Evaluation Results of Selection Bias

All accuracy results after answer-moving attack can be found in Table 9, 10, 11, 12, 13, 14.

H Proof of The Relationship Between MCSB Capability and Selection Bias

For a single instance, we denote its PPA as v_{ppa}, its selection bias as v_{bias}, its accuracy as v_{Acc}. From Equation 1, $v_{\text{bias}} = \frac{\sum_{i=1}^{K}(|v_{\text{Acc}_i} - v_{\text{Acc}_0}|)}{K}$. Considering that v_{Acc_0} can only be 0 or 1 for a single instance, then v_{bias} is:

$$v_{\text{bias}} = \begin{cases} \frac{\sum_{i=1}^{K} v_{\text{Acc}_i}}{K} & v_{\text{Acc}_0 = 0} \\ 1 - \frac{\sum_{i=1}^{K} v_{\text{Acc}_i}}{K} & v_{\text{Acc}_0 = 1} \end{cases} \tag{9}$$

On the other hand, $v_{\text{ppa}} = \frac{\max\limits_{k \sim K}(\sum\limits_{j=1}^{K!} y_j = o_k)}{K!}$. Given a question with K options, there are $K!$ distinct arrangements of these options, we now use a subset, always moving the golden option content to a specific symbol, to represent the overall distribution, then $v_{\text{ppa}} = \frac{\max\limits_{k \sim K}(\sum\limits_{j=1}^{K} y_j = o_k)}{K}$. If the most frequently predicted option is the correct answer, v_{Acc_0} will have a probability of v_{ppa} to take the value of 1, and $v_{\text{ppa}} = \frac{\sum_{i=1}^{K} v_{\text{Acc}_i}}{K}$; else, $v_{\text{ppa}} = \frac{\sum_{i=1}^{K}(1 - v_{\text{Acc}_i})}{K}$. Thus we define:

$$v_{\text{ppa}} = \begin{cases} \frac{\sum_{i=1}^{K} v_{\text{Acc}_i}}{K} & v_{\text{Acc}_0 = 1} \\ 1 - \frac{\sum_{i=1}^{K} v_{\text{Acc}_i}}{K} & v_{\text{Acc}_0 = 0} \end{cases} \tag{10}$$

From Equation 9, 10, we can observe that v_{bias} and v_{ppa} are inversely related with a probability of v_{ppa}. Generally, after SFT, LLMs tend to have higher v_{ppa}, Therefore providing theoretical support for our work.

Benchmarks			**#Samples**	**#Options**	**Golden Answer Distribution**
MMLU	train	-	99842	4	22.2%/25.8%/26.9%/25.1%
	test	STEM	3018	4	21.4%/23.8%/25.9%/28.9%
		Social Science	3077		21.7%/23.4%/23.8%/31.1%
		Humanities	4705		24.2%/24.5%/27.1%/24.2%
		Others	3242		23.8%/26.8%/24.4%/25.1%
		Overall	14042		22.3%/24.7%/25.5%/26.9%
CSQA	train		8971	5	19.4%/20.3%/20.1%/20.4%/19.9%
	test		1221		19.6%/20.9%/19.7%/20.6%19.3%

Table 5: Statistics of all benchmarks.

Model	**Method**	**STEM**			**Social Science**			**Humanities**			**Others**			**MMLU**		
		μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑
LLaMA2-7B	PIF_{raw}	9.6	67.4	43.5	6.2	78.8	61.4	10.1	70.1	48.7	4.9	79.0	59.2	7.9	73.5	52.8
	$\text{PIF}_{\text{Symbol}}$	8.3	70.6	44.1	2.6	83.6	**63.4**	4.7	82.9	51.8	4.5	81.6	**61.1**	3.7	79.6	54.8
	PIF_{SCB}	6.9	68.4	41.5	3.6	81.0	60.4	**2.6**	82.4	49.0	6.1	78.8	59.0	4.4	78.3	52.2
	PIF_{RSCB}	**3.7**	**72.3**	**44.0**	**2.3**	**83.9**	62.7	3.9	**83.1**	**53.1**	**1.3**	**82.9**	60.9	**2.7**	**80.8**	**55.0**
LLaMA2-13B	PIF_{raw}	5.1	73.4	47.9	2.3	84.7	69.4	7.2	78.1	54.2	2.2	84.5	65.2	3.6	80.1	58.7
	$\text{PIF}_{\text{Symbol}}$	**3.2**	**75.1**	46.2	4.0	83.6	66.6	4.9	78.6	56.2	3.6	83.9	62.7	6.5	80.4	57.8
	PIF_{SCB}	4.5	73.6	47.5	2.2	84.2	68.0	4.7	80.3	58.0	**2.1**	**84.9**	64.7	3.2	81.0	59.5
	PIF_{RSCB}	4.8	74.5	**48.0**	**2.0**	**85.1**	**68.5**	**2.9**	**80.9**	**58.4**	2.6	84.2	**65.5**	**3.0**	**81.0**	**60.0**

Table 6: Ablation study on π_{PIF} parameter initialization. PIF_{raw} refers to the model π_{PIF} which is initialized from the models listed in Appendix B. $\text{PIF}_{\text{Symbol}}$, PIF_{SCB}, PIF_{RSCB} refers to the model π_{PIF} which is initialized from π_{Symbol}, π_{SCB}, π_{RSCB} respectively.

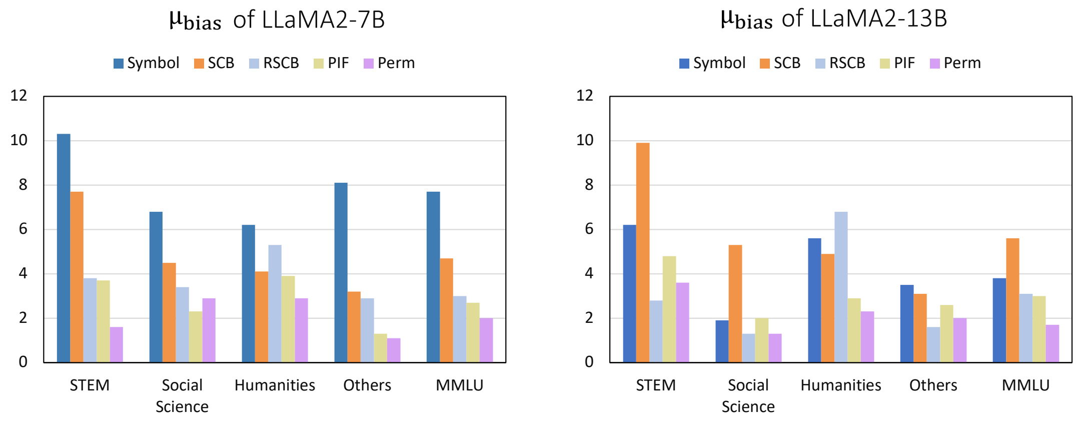


Figure 5: Selection bias of Perm.

Model	**Method**	**STEM**		**Social Science**		**Humanities**		**Others**		**MMLU**	
		μ_{bias}↓	Acc↑	μ_{bias}↓	Acc↑	μ_{bias}↓	Acc↑	μ_{bias}↓	Acc↑	μ_{bias}↓	Acc↑
LLaMA2-7B	Argum.	**2.0**	**44.1**	2.8	62.7	5.9	51.1	1.5	60.4	2.7	54.3
	PIF	3.7	44.0	**2.3**	**62.7**	**3.9**	**53.1**	**1.3**	**60.9**	**2.7**	**55.0**

Table 7: PIF *vs.* Data Argumentation.

Model	**Method**	**STEM**			**Social Science**			**Humanities**			**Others**			**MMLU**		
		μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑	μ_{bias}↓	μ_{ppa}↑	Acc↑
LLaMA2-7B	DPO	8.9	70.6	43.1	3.0	80.4	62.6	3.9	77.2	52.8	5.1	81.6	59.9	5.0	77.5	54.5
	PIF	**3.7**	**72.3**	**44.0**	**2.3**	**83.9**	**62.7**	**3.9**	**83.1**	**53.1**	**1.3**	**82.9**	**60.9**	**2.7**	**80.8**	**55.0**
LLaMA2-13B	DPO	7.6	69.6	47.4	3.8	83.1	67.0	7.5	78.3	58.4	5.5	81.6	63.9	4.9	78.2	58.4
	PIF	**4.8**	**74.5**	**48.0**	**2.0**	**85.1**	**68.5**	**2.9**	**80.9**	**58.4**	**2.6**	**84.2**	**65.5**	**3.0**	**81.0**	**60.0**

Table 8: Point-wise *vs.* Pair-wise.

Move Golden to		orig	A	B	C	D
LLaMA2-7B	Symbol	43.8	61.6 (+17.8)	33.9 (-9.9)	47.2 (+3.4)	33.5 (-10.3)
	SCB	43.6	35.0 (-8.6)	35.4 (-8.2)	52.4 (+8.8)	48.7 (+5.1)
	RSCB	43.2	47.6 (+4.4)	40.7 (-2.5)	39.0 (-4.2)	47.4 (+4.2)
	PIF	44.0	39.5 (-4.5)	45.8 (+1.8)	40.6 (-3.4)	49.2 (+5.2)
LLaMA2-13B	Symbol	46.9	43.4 (-3.5)	57.4 (+10.5)	42.3 (-4.6)	53.4 (+6.5)
	SCB	48.7	36.2 (-12.5)	41.9 (-6.8)	45.2 (-3.5)	65.7 (+17.0)
	RSCB	47.5	47.3 (-0.2)	51.2 (+3.7)	49.2 (+1.7)	41.8 (-5.7)
	PIF	48.0	47.5 (-0.5)	45.7 (-2.3)	58.8 (+10.8)	42.4 (-5.6)

Table 9: The accuracy results after answer-moving attack on the LLMs with STEM benchmarks.

Move Golden to		orig	A	B	C	D
LLaMA2-7B	Symbol	60.6	73.8 (+13.2)	51.6 (-9.0)	64.2 (+3.6)	53.9 (-6.7)
	SCB	60.4	58.5 (-1.9)	54.9 (-5.5)	63.6 (+3.2)	62.7 (+2.3)
	RSCB	60.5	62.1 (+1.6)	58.9 (-1.6)	56.8 (-3.7)	65.0 (+4.5)
	PIF	60.9	59.3 (-1.6)	60.7 (-0.2)	59.3 (-1.6)	62.9 (+2.0)
LLaMA2-13B	Symbol	65.5	64.3 (-1.2)	70.5 (+5.0)	59.9 (-5.6)	63.4 (-2.1)
	SCB	64.8	62.1 (-2.7)	63.8 (-1.0)	61.8 (-3.0)	70.5 (+5.7)
	RSCB	64.6	65.0 (+0.4)	66.3 (+1.7)	66.0 (+1.4)	61.8 (-2.8)
	PIF	65.6	64.7 (-0.9)	63.8 (-1.8)	70.1 (+4.5)	62.0 (-3.6)

Table 10: The accuracy results after answer-moving attack on the LLMs with Others benchmarks.

Move Golden to		orig	A	B	C	D
LLaMA2-7B	Symbol	62.5	73.0 (+10.5)	54.5 (-8.0)	64.6 (+2.1)	55.9 (-6.6)
	SCB	62.1	56.8 (-5.3)	55.9 (-6.2)	65.2 (+3.1)	65.4 (+3.3)
	RSCB	62.3	61.2 (-1.1)	59.4 (-2.9)	56.8 (-5.5)	66.6 (+4.3)
	PIF	62.7	59.5 (-3.2)	60.5 (-2.2)	61.1 (-1.6)	65.2 (+2.5)
LLaMA2-13B	Symbol	68.1	66.7 (-1.4)	70.0 (+1.9)	64.0 (-4.1)	67.8 (-0.3)
	SCB	68.5	62.0 (-6.5)	64.7 (-3.8)	63.6 (-4.9)	74.7 (+6.2)
	RSCB	68.0	65.2 (-2.8)	67.8 (-0.2)	69.9 (+1.9)	67.8 (-0.2)
	PIF	68.5	67.3 (-1.2)	65.8 (-2.7)	70.0 (+1.5)	65.8 (-2.7)

Table 11: The accuracy results after answer-moving attack on the LLMs with Social science benchmarks.

Move Golden to		orig	A	B	C	D
LLaMA2-7B	Symbol	52.2	58.0 (+5.8)	43.3 (-8.9)	57.4 (+5.2)	47.7 (-4.5)
	SCB	50.5	45.9 (-4.6)	45.4 (-5.1)	54.0 (+3.5)	53.7 (+3.2)
	RSCB	52.1	46.1 (-6.0)	54.0 (+1.9)	45.5 (-6.6)	58.4 (+6.3)
	PIF	53.1	42.4 (-10.7)	53.0 (-0.1)	51.5 (-1.6)	56.7 (+3.6)
LLaMA2-13B	Symbol	55.0	47.9 (-7.1)	61.6 (+6.6)	57.4 (+2.4)	61.2 (+6.2)
	SCB	54.2	43.3 (-10.9)	55.8 (+1.6)	55.9 (+1.7)	65.5 (+11.3)
	RSCB	55.8	44.3 (-11.5)	61.1 (+5.3)	64.3 (+8.5)	57.9 (+2.1)
	PIF	58.4	52.2 (-6.2)	58.9 (+0.5)	61.6 (+3.2)	56.6 (-1.8)

Table 12: The accuracy results after answer-moving attack on the LLMs with Humanities benchmarks.

Move Golden to		orig	A	B	C	D
LLaMA2-7B	Symbol	54.6	65.7 (+11.1)	45.6 (-9.0)	58.4 (+3.8)	47.8 (-6.8)
	SCB	53.8	48.9 (-4.9)	47.7 (-6.1)	58.3 (+4.5)	57.2 (+3.4)
	RSCB	54.4	53.5 (-0.9)	53.5 (-0.9)	49.0 (-5.4)	59.3 (+4.9)
	PIF	55.0	49.5 (-5.5	54.9 (-0.1)	53.0 (-2.0)	58.4 (+3.4)
LLaMA2-13B	Symbol	59.2	54.8 (-4.4)	64.6 (+5.4)	56.3 (-2.9)	61.5 (+2.3)
	SCB	58.6	50.2 (-8.4)	56.6 (-2.0)	56.7 (-1.9)	68.7 (+10.1)
	RSCB	58.7	54.3 (-4.4)	61.7 (+3.0)	62.7 (+4.0)	57.5 (-1.2)
	PIF	60.0	57.4 (-2.6)	58.7 (-1.3)	64.8 (+4.8)	56.8 (-3.2)

Table 13: The accuracy results after answer-moving attack on the LLMs with MMLU benchmarks.

Move Golden to		orig	A	B	C	D	E
LLaMA2-7B	Symbol	79.8	82.6 (+2.8)	77.2 (-2.6)	80.7 (+0.9)	79.1 (-0.7)	79.6 (-0.2)
	SCB	78.4	76.2 (-2.2)	77.1 (-1.3)	80.3 (+1.9)	79.2 (+0.8)	82.3 (+3.9)
	RSCB	79.7	82.8 (+3.1)	81.8 (+2.1)	79.9 (+0.2)	74.9 (-4.8)	77.8 (-1.9)
	PIF	79.4	79.8 (+0.4)	77.6 (-1.8)	81.7 (+2.3)	77.3 (-2.1)	77.8 (-1.6)
LLaMA2-13B	Symbol	81.5	81.3 (-0.2)	76.2 (-5.3)	86.2 (+4.7)	82.9 (+1.4)	83.7 (+2.2)
	SCB	81.0	79.0 (-2.0)	81.8 (+0.8)	82.5 (+1.5)	87.6 (+6.6)	79.1 (-1.9)
	RSCB	80.9	81.8 (+0.9)	82.9 (+2.0)	82.5 (+1.6)	77.9 (-3.0)	78.8 (-2.1)
	PIF	82.2	79.9 (-2.3)	82.0 (-0.2)	82.5 (+0.3)	83.4 (+1.2)	81.7 (-0.5)

Table 14: The accuracy results after answer-moving attack on the LLMs with CSQA benchmarks.

Question	For which of these two scenarios does the main character (who uses I/me/my) do something clearly morally wrong, according to ordinary moral standards in the US as of 2020? Scenario 1 | I slammed on my breaks to miss the ball as it came in to the road. Scenario 2 | I taught my children to play the xylophone.
Options	A:Wrong, Wrong B:Wrong, Not wrong C:Not wrong, Wrong D:Not wrong, Not wrong
Option Symbols	s_1: A s_2: B s_3: C s_4: D
Option contents	o_1: Wrong, Wrong o_2: Wrong, Not wrong o_3: Not wrong, Wrong o_4: Not wrong, Not wrong
Arrangements	a_1: A: Wrong, Wrong B: Wrong, Not wrong C: Not wrong, Wrong D: Not wrong, Not wrong a_2: A: Wrong, Wrong B: Wrong, Not wrong C: Not wrong, Not wrong D: Not wrong, Wrong a_3: A: Wrong, Wrong B: Not wrong, Wrong C: Wrong, Not wrong D: Not wrong, Not wrong a_4: A: Wrong, Wrong B: Not wrong, Wrong C: Not wrong, Not wrong D: Wrong, Not wrong a_5: A: Wrong, Wrong B: Not wrong, Not wrong C: Wrong, Not wrong D: Not wrong, Wrong a_6: A: Wrong, Wrong B: Not wrong, Not wrong C: Not wrong, Wrong D: Wrong, Not wrong a_7: A: Wrong, Not wrong B: Wrong, Wrong C: Not wrong, Wrong D: Not wrong, Not wrong a_8: A: Wrong, Not wrong B: Wrong, Wrong C: Not wrong, Not wrong D: Not wrong, Wrong a_9: A: Wrong, Not wrong B: Not wrong, Wrong C: Wrong, Wrong D: Not wrong, Not wrong a_{10}: A: Wrong, Not wrong B: Not wrong, Wrong C: Not wrong, Not wrong D: Wrong, Wrong a_{11}: A: Wrong, Not wrong B: Not wrong, Not wrong C: Wrong, Wrong D: Not wrong, Wrong a_{12}: A: Wrong, Not wrong B: Not wrong, Not wrong C: Not wrong, Wrong D: Wrong, Wrong a_{13}: A: Not wrong, Wrong B: Wrong, Wrong C: Wrong, Not wrong D: Not wrong, Not wrong a_{14}: A: Not wrong, Wrong B: Wrong, Wrong C: Not wrong, Not wrong D: Wrong, Not wrong a_{15}: A: Not wrong, Wrong B: Wrong, Not wrong C: Wrong, Wrong D: Not wrong, Not wrong a_{16}: A: Not wrong, Wrong B: Wrong, Not wrong C: Not wrong, Not wrong D: Wrong, Wrong a_{17}: A: Not wrong, Wrong B: Not wrong, Not wrong C: Wrong, Wrong D: Wrong, Not wrong a_{18}: A: Not wrong, Wrong B: Not wrong, Not wrong C: Wrong, Not wrong D: Wrong, Wrong a_{19}: A: Not wrong, Not wrong B: Wrong, Wrong C: Wrong, Not wrong D: Not wrong, Wrong a_{20}: A: Not wrong, Not wrong B: Wrong, Wrong C: Not wrong, Wrong D: Wrong, Not wrong a_{21}: A: Not wrong, Not wrong B: Wrong, Not wrong C: Wrong, Wrong D: Not wrong, Wrong a_{22}: A: Not wrong, Not wrong B: Wrong, Not wrong C: Not wrong, Wrong D: Wrong, Wrong a_{23}: A: Not wrong, Not wrong B: Not wrong, Wrong C: Wrong, Wrong D: Wrong, Not wrong a_{24}: A: Not wrong, Not wrong B: Not wrong, Wrong C: Wrong, Not wrong D: Wrong, Wrong
Model outputs	y_1: Wrong, Not wrong y_2: Wrong, Not wrong y_3: Not wrong, Wrong y_4: Not wrong, Wrong y_5: Wrong, Not wrong y_6: Wrong, Not wrong y_7: Wrong, Wrong y_8: Wrong, Wrong y_9: Not wrong, Wrong y_{10}: Wrong, Wrong y_{11}: Not wrong, Wrong y_{12}: Wrong, Wrong y_{13}: Wrong, Wrong y_{14}: Wrong, Not wrong y_{15}: Wrong, Not wrong y_{16}: Wrong, Not wrong y_{17}: Wrong, Wrong y_{18}: Wrong, Wrong y_{19}: Wrong, Not wrong y_{20}: Wrong, Wrong y_{21}: Wrong, Not wrong y_{22}: Wrong, Not wrong y_{23}: Not wrong, Wrong y_{24}: Wrong, Not wrong
Frequency	o_1(Wrong, Wrong): $8 \div 4! \approx 0.333$ o_2(Wrong, Not wrong): $11 \div 4! \approx 0.458$ o_3(Not wrong, Wrong): $5 \div 4! \approx 0.208$ o_4(Not wrong, Not wrong): $0 \div 4! = 0$
PPA result	0.458

Table 15: An example of the PPA calculation process for a single instance. Arrangements are the results of the arrangement of option contents. Model output y_i refers to the answer produced by the model after the question and a_i are input into it. The PPA metric does not consider whether the model performs tasks rightly. It measures the consistency of the model during the execution of tasks.