Eliminating Positional Bias in LLMs via Attention Weight Averaging

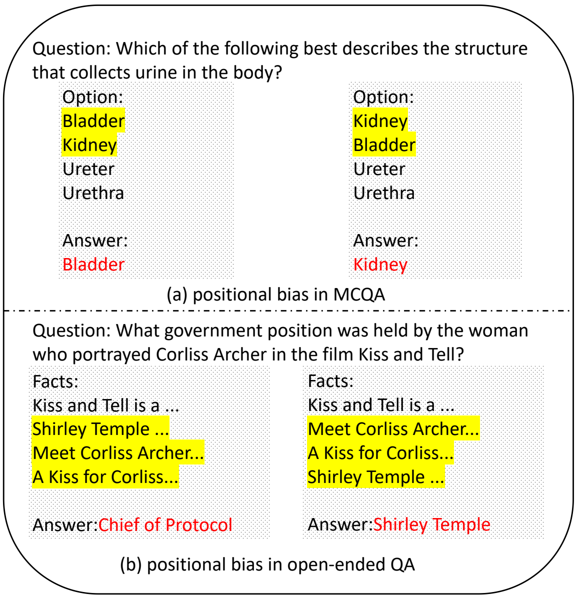
Anonymous ACL submission

Abstract

Positional bias in LLMs means that changing the order of input sentences leads to semantic inconsistency in the output. Positional bias occurs even though the overall meaning of the input remains the same. Recent studies have observed and verified that positional bias is prevalent across various LLMs and tasks. Our study proposes the Average Attention Infer module, which starts from the calculation of the attention mechanism and aims to reduce positional bias by computing the average attention weight of different arrangements. We design experiments to verify the module's effectiveness in mitigating positional bias. It is also verified that the LLMs can still maintain their language functions after debiasing, which makes our module easy to extend to other tasks. Methods for selecting layers and permutations are provided to accelerate the module's computation further. We release the code[1] and hope this research can inspire the design and research of a new generation of attention modules, thereby contributing to the fundamental elimination of positional bias.

1 Introduction

Positional bias in large language models (LLMs) can be interpreted differently depending on the context. In the MCQA setting, Wang et al. (2023) interprets it as the model's inherent preference for certain positional options. The issue we study, however, is a type of positional bias that occurs in text generation models. Figure 1 specifically illustrates the meaning of this kind of positional bias. Simply put, positional bias, which is studied in our paper, refers to the phenomenon where the semantic output of the model changes significantly, even though there are changes in position but minimal changes in semantics in the model's input.



Question: Which of the following best describes the structure that collects urine in the body?

Option:
Bladder
Kidney
Ureter
Urethra

Answer:
Bladder

Option:
Kidney
Bladder
Ureter
Urethra

Answer:
Kidney

(a) positional bias in MCQA

Question: What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?

Facts:
Kiss and Tell is a ...
Shirley Temple ...
Meet Corliss Archer...
A Kiss for Corliss...

Answer:Chief of Protocol

Facts:
Kiss and Tell is a ...
Meet Corliss Archer...
A Kiss for Corliss...
Shirley Temple ...

Answer:Shirley Temple

(b) positional bias in open-ended QA

Figure 1: Positional bias in MCQA and open-ended QA. Position-changed words are marked in yellow.

The existence of positional bias significantly undermines the usability of LLMs in various domains. For example, in the evaluation task, which uses LLMs to compare and evaluate multiple candidates, and in multi-condition question answering, which requires LLMs to reason and respond based on given conditions. Positional bias is frequently studied in the context of multiple-choice question answering (MCQA) because MCQA naturally contains commutative parts. The order of options in MCQA is generally considered not to influence the final result, as stated in Wang et al. (2024a), *A LLM can be deemed proficient in answering a specific MCQA-format question only if it consistently predicts the same correct answer across all permutations of option orders.* This should also be followed in open-ended QA tasks (Chen et al., 2024), where answers of LLMs can not be classified by options in MCQA. Positional bias is an inherent robustness issue within LLMs that needs

[1] Code and results are available at https://anonymous.4open.science/r/Average-Attention-Infer-BD1D/.

to be addressed.

Contemporary mainstream LLMs are predominantly based on the attention mechanism of transformer architecture (Vaswani et al., 2017; Jiang et al., 2023; Touvron et al., 2023). Attention computation constitutes a significant portion of the overall calculation of output logits. During the attention computation process, the attention weight is often interpreted as the degree to which the model focuses on different parts of the input text (Shin et al., 2024; Hao et al., 2021; Voita et al., 2019). Intuitively, one can hypothesize that if LLMs correctly understand the commutative part, the attention weight on different options should shift correspondingly with their positions.

We test the attention weights of the same option in different permutations and observe that the relative magnitude of the attention weights of tokens following the options does not shift accordingly with the permutation changes. Therefore, it is reasonable to believe that one possible cause of positional bias is that the attention weights of the same option change with different permutations.

Based on this hypothesis, we propose a debiasing module called Average Attention Infer (AAT) to compute unbiased attention weights for each option, thereby internally eliminating the positional bias introduced by the attention mechanism. We demonstrate that AAT achieves superior debiasing effectiveness to strong baselines, especially in small models with fewer permutations. We further test AAT on open-ended QA tasks. The results confirm that our approach works well and has minimal detrimental effects on language abilities,

We further investigate the impact of selecting different layers and permutation sets on the effectiveness of AAT and make trade-offs between performance and latency. We observe that for different models, only certain layers are order-sensitive.

Summary of Contributions (1) We observed and validated that the variation of attention weights with permutations is a major cause of positional bias; (2) We proposed a training-free plugin module to correct positional bias in LLMs with attention mechanisms, achieving significant effectiveness; (3) Our experiments revealed that positional bias is model-specific and relatively stable, leading to the development of a more efficient module based on this insight.

2 Method

We formulate the problem of positional bias in LLMs and introduce our method in this part.

2.1 Problem Formulation

In this section, we provide a definition of the positional bias in LLM. We make the definition as general as possible to fit it into more tasks.

Question with commutative part Positional bias is significant only if the input contains some parts whose permutations make little difference to the ground truth answer. It's similar to the commutative property in math, so we name the part as *commutative part*.

C is a composition of sentences $C = c_1c_2c_3...c_n \subseteq Q$, Q is the question. c_i is called a commutative unit. $\mathcal{A}_Q = Answer(Q)$ represents the set of all ground truth answer of Q. $\mathcal{I}$ is the set of full permutations of $\{1, 2, 3, ..., n\}$.

Definition 2.1 (commutative part) $\forall I \in \mathcal{I}$, *the corresponding part* C_I *satisfy* ***commutative term*** $\frac{|\mathcal{A}_{Q_{C_I}} \cap \mathcal{A}_{Q_C}|}{|\mathcal{A}_{Q_{C_I}} \cup \mathcal{A}_{Q_C}|} \geq \epsilon$, *then* C *is called an* ϵ*-commutative part, noted as* $C \in \mathcal{C}_\epsilon(Q)$. $\epsilon \in [0, 1]$ *is the threshold.*

As the ϵ is closer to 1, the constraint is more strict. Questions with commutative parts can be defined as questions that hold one or more commutative parts.

One may wonder why we give such an abstract ϵ rather than restricting the space of ground truth answers to the same one. This setting is motivated by the consistency of semantic space. The set of ground truth answers can only be partially unchangeable under the open-ended QA setting where the model gives answers from the whole semantic space. Thus, there should always be this kind of ground truth answers that *My answer is apple, the third answer.* and *My answer is apple, the second answer.* for different permutations. The two answers are not the same in semantic space, so we can't say that the two ground truth semantic spaces of permutations are the same. It means that $\epsilon = 1$ is almost impossible.

Another concern is how to set the value of ϵ. As we mentioned above, what we expect is that ground truth answers without information about the position should belong to both $\mathcal{A}_{Q_{C_{I_1}}}$ and $\mathcal{A}_{Q_{C_{I_2}}}$. However, estimating ϵ in the infinite semantic space is hard. The threshold is defined for generalisation.

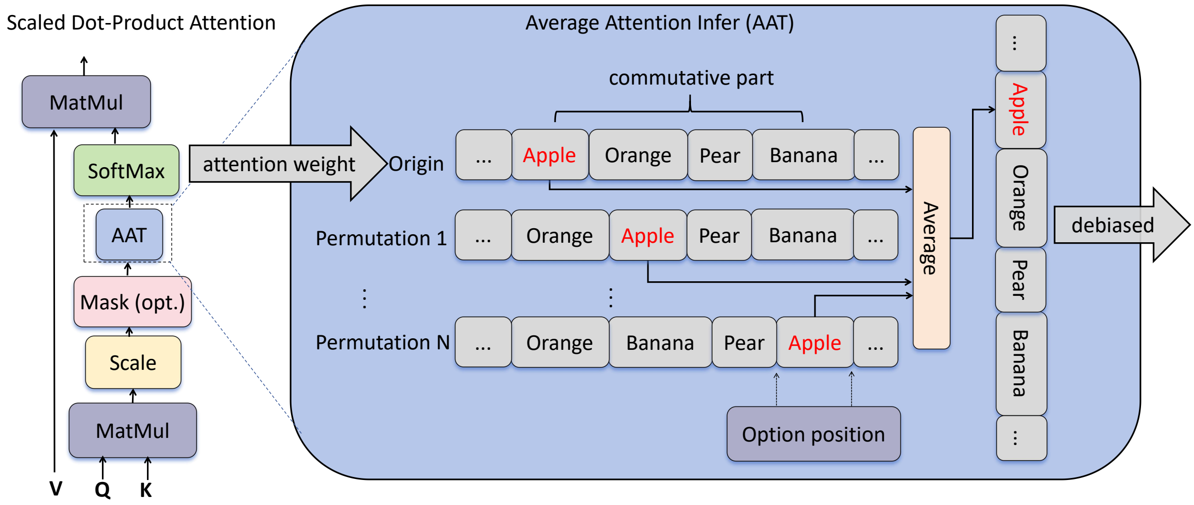


Figure 2: The framework of Average Attention Infer (AAT); AAT will first generate the permutation according to option position input and align the attention weight on semantic granularity.

We also cancel the symbol of option in MCQA to avoid the token bias, which can be explained by the same pattern as before. This paper's following studies about MCQA are all in the no-symbol setting.

The **commutative term** can be different in practice. While we define it as a kind of Intersection over Union (IoU), one can set different semantic similarity functions for different purposes. For MCQA, the option part can be naturally treated as *commutative part* after removing cases including options like *None of above choices* and *A and B*. The semantic space of answers can be classed into option labels.

Positional bias We then introduce the positional bias in questions holding commutative parts. The positional bias intuitively indicates the situation that when we replace the $\mathcal{A}(Q)$ with $M(Q)$ of a model M, it will break the definition of *commutative part*.

To make positional bias more practical, we introduce a label function $Label : \mathbb{R}^d \rightarrow \mathbb{R}$ to map an answer A to a class. Now, we can give a simplified but practical version of positional bias.

Definition 2.2 (positional bias) $\exists I, J \in \mathcal{I}_{Q_C} \wedge I \neq J \rightarrow Label(M(Q_{C_I})) \neq Label(M(Q_{C_J}))$

$\mathcal{I}_{Q_C}$ is the full permutations set of commutative part $C \subseteq Q$. We provide two main metrics to evaluate a model's performance on a dataset D from the perspective of positional bias. We will first test M on the full permutations of D and evaluate the results.

The first is called the top-k vector. It's the expected distribution of the label proportion in descending order among one full permutation batch for all cases in the dataset.

Definition 2.3 (top-k vector)

$$V^i_{top-k} = Desc(\sum_{I \in \mathcal{I}_{Q^i_C}} Label(M(Q^i_{C_I}))) \quad (1)$$

$$V_{top-k} = Softmax(E_{i \in D}(V^i_{top-k})) \quad (2)$$

$Desc$ indicates the descending sort function, and the $Label$ function in top-k should be in one-hot encoding format. The top-k vector only cares about how consistent the model's answers are in a batch (full permutation of one case). Intuitively, $\sum V_{top-k} = 1$. The theoretical upper bound of top-k is a zero-like vector except for the first position.

The second is called permutation invariant ratio, indicating the proportion of examples in a dataset D where model M shows no positional bias on them.

Definition 2.4 (PIR)

$$PIR = \frac{\sum_i^{|D|} \mathbb{I}(V^i_{top-k}[2] = 0)}{|D|} \quad (3)$$

$\mathbb{I}$ is the indicator function and $V^i_{top-k}[2]$ represents the top-2 value of top-k vector. When the top-2 value is 0, the top-k vector achieves the optimal $[1, 0, 0, ..., 0]$. This also indicates that the model's answers to permutations of one case are labelled into the same class. The upper bound of PIR is 1 without doubt.

2.2 Attention Weight

In this part, we introduce our motivation and the important finding about attention weight. LLMs based on transformer can be modelled as a sequence of some transformer blocks. Every block calculates an attention weight. Concretely, $A_w = Softmax(QK^T)$. A_w refers to attention weight. Attention is derived from A_w and the value matrix then. After this, attention should be added to the residual and normalised as the input of the feed-forward layer. In the classic scaled dot-product attention, A_w is often treated as how much attention the model pays to other tokens in some studies of interpretability about LLMs (Vashishth et al., 2019; Serrano and Smith, 2019; Mrini et al., 2020). An intuitive thought is that *despite the permutation, LLMs should pay the same proportion of attention weight among all commutative units*.

To verify if attention weights change as expect, we test a random case from CosmosQA (Huang et al., 2019) on Llama2-13b-chat (Touvron et al., 2023).

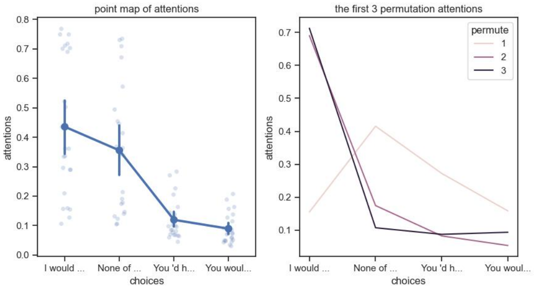


Figure 3: Attention weights of the full permutation of one case in CosmosQA, forwarded by Llama2-13b-chat; sum on all layers and all heads.

Figure 3 shows attention weights of different choices among different permutations in one case. Attention weights are forwarded through a softmax function among all four options. The left in Figure 3 shows that option *'I would ...'* and *'None of ...'* get dynamic attention weights which vary from 0.75 to 0.1. The right part of Figure 3 indicates that in permutation 1, the model pays the most attention to option *'None of ...'* while in permutations 2 and 3, the option *'I would ..'* is the focused one.

According to the results above, attention weights sometimes change differently. The softmax value of an option's attention weight changes as the permutation changes. This demonstrates that positional bias has been introduced when we calculate attention weights.

2.3 Average Attention Infer

We then propose our method, Average Attention Infer (AAT for short). Figure 2 demonstrates AAT. The method is straightforward. AAT gets an external input called option's position, generally speaking, positions of commutative units. AAT then averages attention weights of these positions on semantic granularity and replaces them with the corresponding averaged values.

Algorithm 1 Attention Weight Alignment

```
Input: $W_A$, $Pos$, $Perm$
Output: $W_A^d$
$N_c = len(Pos[1])$, $Bsz = len(W_A)$
$S_A = [0] * N_c$
for each $i \in [1, Bsz]$ do
    for each $j \in [1, N_c]$ do
        $S_A[Perm[i][j]] += W_A[i][Pos[i][j]]$
    end for
end for
$S_A = S_A / Bsz$
$W_A^d = FillAtt(S_A, W_A, Pos, Perm)$
return $W_A^d$
```

Algorithm 1 shows how we calculate the debiased attention weight. W_A refers to attention weights of a batch of full permutations of one question. Pos is the position of commutative units in each permutation of the batch. $Perm$ contains the information of every permutation in the batch. The corresponding index is related to the origin order of every commutative unit. Outputs are debiased attention weights attending the following calculation.

The latency mainly comes from external calculations on different permutations and several layers. Obviously, the additional time is $CN_PN_LO(n)$ for a question with n commutative parts, where C is the number of commutative units in every commutative part (suppose they are the same), N_P and N_L are the size of the permutation set and the number of layers. For every single layer, the time complexity of Algorithm 1 is $CN_PO(n)$.

We mainly try three kinds of permutation sets. k refers to the number of commutative units.

single permutation Choose one permutation as the anchored permutation (always the origin permutation) and apply it to all other permutations. $\mathcal{I}_{Q_C} = \{(1, 2, 3, 4, ..., k)\}$

cyclic permutation Following the work in Zheng et al. (2024), the cyclic permutation is chosen as below. Every commutative unit shows up at every position only once.

$\mathcal{I}_{Q_C} = \{(i, i+1, ..., k, 1, ..., i-1)\}_{i=1}^{k}$

full permutation The full permutation.

3 Experiment

In this section, we will first test the effectiveness of AAT on multiple transformer-based open-source LLMs, as well as on two multiple-choice question answering (MCQA) datasets, MMLU (Hendrycks et al., 2020) and CosmosQA (Huang et al., 2019), to verify AAT's ability to eliminate positional bias. The metrics include accuracy, PIR and top-1 value. In order to reduce the time consumed by AAT, we investigate the impact of different layers and permutations on AAT. To assess the disruptive impact of our method on text, we test the effectiveness of AAT in the context of text-based multiple-choice question answering (MCQA). Finally, we demonstrate how to adapt AAT to open-ended question-answering tasks and evaluate its effectiveness.

All tests are conducted in a 0-shot setting. Additionally, to eliminate bias caused by option labelling, all tests are carried out with labels removed from the options. More details can be found in Appendix C.

3.1 Debiasing Results

We first test AAT on all layers and the full permutations. Results are shown in Table 1. Since top-1 and PIR are used to measure the consistency of a model across different permutations of questions, the results of majority vote on both top-1 and PIR metrics are guaranteed to achieve the theoretical optimal value of 1. This is because the majority vote directly assigns a single answer to the entire set of permutations. Even if the assignment is made randomly, the responses are guaranteed to be consistent.

According to the results in Table 1, we find that the proposed method effectively enhances both PIR and top-1 values, thereby also achieving an increase in ACC. The test results across multiple datasets and models approach the theoretical optimal value of 1. This confirms our hypothesis that **bias in attention weights is one of the primary sources of positional bias**. Consequently, it validates the effectiveness of AAT.

The test results on ChatGLM3 indicate that the Acc of AAT even surpasses that of the majority vote. This suggests that AAT may have a significant advantage over the majority vote when the model size is smaller or when the model's accuracy is low.

Both the majority vote and AAT methods enhance the original model's accuracy. To explore the inner difference between these two methods, we additionally calculate *average difference*, representing the proportion of debiased answers by the method, along with the proportions of three types of modifications (T->F, F->F, F->T). Due to space limitations, we only show results on CosmosQA. Results on MMLU are in Appendix D.1.

The results in Table 2 indicate that AAT's advantage over the majority vote lies in its ability to perform bias reduction when the permutation is relatively small. In contrast, the effectiveness of the majority vote improves as the number of permutations increases. However, the majority vote will introduce more latency than AAT as AAT only computes several attention weights, while the majority vote needs to go through the whole network.

For smaller-sized models, such as the ChatGLM3-6b, AAT outperforms the majority vote across all permutations. In summary, AAT generally outperforms the majority vote method when used with smaller permutations and models with smaller parameter sizes. Conversely, majority vote has advantages on larger models and when permutations involve full permutations. More experiments about model size are shown in Appendix D.

3.2 Layer and Permutation

Although ATT achieves commendable results in eliminating positional bias, its substantial cost limits its applicability. The additional cost mainly comes from the permutation size and the calculation on every layer. We test on different layers and permutations to study how these two factors make differences in AAT. Only the heatmaps of Llama2-13b-chat's ΔPIR are reported below due to the space. The full results are in Appendix D.

Since Llama2 and Qwen1.5 both have 40 layers, we conducted sliding tests with a step size of 5 and a window size of 10. For ChatGLM3, which has 28 layers, we set both the step size and window size to 4. We choose 6 different permutation sizes. 1, 4 and 24 are the same as before. Other permutations are formed by one original permutation and $n-1$ random permutations from the full permutation set.

Model	Method	CosmosQA			MMLU		
		Acc	Top-1	PIR	Acc	Top-1	PIR
Llama2 (Touvron et al., 2023)	origin	0.5780	0.8119	0.3010	0.5109	0.7753	0.3140
	MV	**0.6211**	1.0000	1.0000	**0.5526**	1.0000	1.0000
	AAT	0.5906	0.9691	0.8729	0.5461	0.9383	0.7579
	AAT+MV	0.6020	1.0000	1.0000	0.5456	1.0000	1.0000
	AAT EF	0.5994	0.9472	0.7860	0.5278	0.9183	0.6912
ChatGLM3 (Du et al., 2022)	origin	0.4801	0.7504	0.2274	0.4804	0.7694	0.2684
	MV	0.5284	1.0000	1.0000	0.5053	1.0000	1.0000
	AAT	0.5471	0.9509	0.7793	0.5203	0.9512	0.8123
	AAT+MV	**0.5552**	1.0000	1.0000	**0.5228**	1.0000	1.0000
	AAT EF	0.5330	0.9144	0.6321	0.4980	0.9296	0.7316
Qwen1.5 (Bai et al., 2023)	origin	0.7251	0.8800	0.5552	0.6341	0.8240	0.4211
	MV	**0.7513**	1.0000	1.0000	**0.6719**	1.0000	1.0000
	AAT	0.7288	0.9627	0.8528	0.6621	0.9560	0.8298
	AAT+MV	0.7291	1.0000	1.0000	0.6702	1.0000	1.0000
	AAT EF	0.7189	0.9606	0.8328	0.6676	0.9624	0.8456

Table 1: Debiasing results on three models and two MCQA datasets. MV refers to majority vote explained in B. The parameter numbers of Llama2, ChatGLM3 and Qwen1.5 are 13b, 6b and 14b. All three models are tested on the chat version. ATT EF 3.2 refers to choosing only efficient layers and average on cyclic permutation. The best values of each model are in bold.

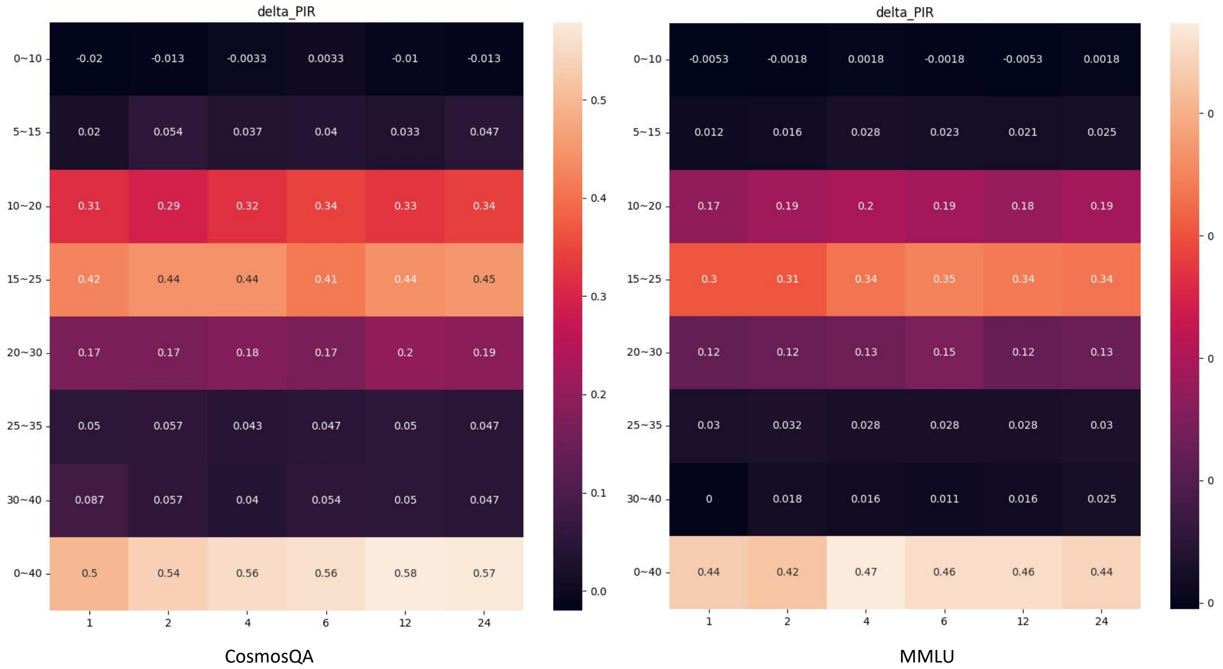


Figure 4: Heatmap of ΔPIR ($\Delta PIR = PIR(debiased) - PIR(origin)$) on different layer and permutation set; the last line indicates the full layer. Only Llama2-13b-chat results are shown. Full results are in Appendix D.

Model	Method	PM	T->F% ↓	F->F(%)	F->T(%)↑	Avg Diff(%)
Llama2	MV	1	40.7	29.5	29.8	27.1
	AAT	1	**36.3**	29.8	**33.8**	24.2
	MV	4	29.0	30.5	40.5	20.9
	AAT	4	**28.6**	29.6	**41.7**	18.3
	MV	24	**24.4**	30.1	45.4	19.4
	AAT	24	24.8	28.0	**47.3**	17.1
ChatGLM3	MV	1	30.5	34.4	35.1	32.3
	AAT	1	**28.9**	35.1	**36.0**	29.8
	MV	4	24.0	35.3	40.7	26.9
	AAT	4	**23.4**	32.8	**43.8**	30.3
	MV	24	24.5	32.5	43.0	26.0
	AAT	24	**21.8**	33.5	**44.6**	29.5
Qwen1.5	MV	1	42.9	26.2	31.0	16.2
	AAT	1	**38.8**	25.4	**35.8**	15.8
	MV	4	**35.9**	27.6	**36.5**	13.2
	AAT	4	36.6	30.8	32.6	15.2
	MV	24	**26.6**	27.2	**46.2**	12.3
	AAT	24	34.3	28.8	36.9	14.3

Table 2: The breakdown results of models' difference from the origin on CosmosQA. PM=1 for single permutation, 4 for cyclic permutation and 24 for full permutation

According to Figure 4, some findings are conducted below.

Only several layers contribute to positional bias in the result. The results from the three models on the two datasets indicate that only certain layers contribute significantly to the final positional bias. This insight prompts us to optimise AAT to target specific layers rather than debiasing across all layers.

Different model holds differently effective layers despite the dataset. The experimental results suggest that different models actually consist of varying effective layers. According to Figure 7, for Llama2, layers between 10 and 30 are primarily responsible for generating positional bias. For Qwen1.5, the first 10 layers appear to be more significant. In the case of ChatGLM, the key layers are concentrated between 16 and 24. Furthermore, by comparing layer characteristics across different datasets, we find that layer features are model-specific. This indicates that we can initially estimate which layers are more crucial for reducing positional bias using a small dataset, thereby achieving high debiasing effectiveness with low latency.

Cyclic permutation is enough. Almost all heatmaps indicate that the benefits of increasing permutations beyond four become marginal, achieving 90% of the effectiveness of a complete permutation on ΔPIR metrics. In some settings, such as Qwen1.5 on the MMLU ΔPIR chart shown in Figure 7, a permutation of four even surpasses the full permutation. This suggests that cyclic permutation is entirely sufficient. However, when permutation is set to two, its performance is inferior to that of cyclic permutation in most settings. This is expected, as cyclic permutation includes one permutation of each option at every position, whereas permutation=2 simply adds one random permutation in addition to the original, conveying significantly less information than cyclic permutations.

Based on the previous analysis, we select specific layers for the three models—layers 10 to 30 for Llama2, layers 0 to 10 for ChatGLM, and layers 16 to 24 for Qwen—and test them with a permutation of 4 on two datasets. The results are presented in row AAT EF of Table 1.

AAT EF achieved 90% of AAT's performance under all settings with less than 4x latency (AAT only needs the permutation's attention weight). AAT EF even surpassed AAT in tests conducted on the Qwen1.5-14b-chat model on the MMLU dataset.

3.3 AAT on Text

We first test AAT on MMLU with text setting, which means we filter the answer from the words generated by LLM. We use the greedy decoding strategy by setting do-sample to false to avoid bias from randomness. Invalid ratio refers to the proportion of invalid answers. Llama2's heatmaps are shown in Figure 8 due to space limitations. According to Figure 8, Llama2-13b-chat introduces less than 5% invalid ratio on layer intervals 15-25 and 20-30. However, on 10-20 layers, 23% invalid ratio is introduced, which inspires us that the best interval of Llama2 may not be 10-30 but 15-30. This also indicates that arbitrarily altering the attention weights of certain layers in a model could greatly affect its usability, even though these layers may not significantly impact the final debiasing effect in a probabilistic setting.

In practice, one should select layers carefully according to several metrics to ensure AAT EF's performance.

Our method can actually be extended to more tasks. This aspect will be tested on the HotpotQA (Yang et al., 2018) dataset to evaluate the debiasing effect of AAT. We filter the facts in the original dataset into only 4 facts and input them as the context. Thus, permutation is built by changing the order of facts. In open-ended question answering, the number of semantic classes is not sure. Thus we simply divide the answer into true and false.

We only detect instances where at least one correct answer is present among the fully permuted responses. These cases are called *potential cases*. Potential permutation invariant ratio (PPIR) refers to the number of instances where all answers are correct divided by the number of potential cases. Results are shown in Table 3. These conclusions remain valid and even perform better in few-shot scenarios. Few-shot results are in Appendix D.

Model	Method	Acc	PPIR
Llama2	origin	0.3242	0.2233
	AAT EF	0.3786	0.4324
ChatGLM3	origin	0.5675	0.3043
	AAT EF	0.5157	0.3853
Qwen1.5	origin	0.7788	0.6177
	AAT EF	0.7828	0.8520

Table 3: AAT EF on open-ended QA task (HotpotQA).

According to the results in Table 3, AAT EF still performs very well under the Qwen1.5 model, but its performance on Llama2 and ChatGLM3 is not as good as on MCQA. This may be due to the length of the commutative parts. The attention weight will finally be softmax, which means every word will get less attention weight than in MCQA. As a result, the debiasing effect of AAT is diluted.

4 Related Work

Attention Interpretability Attention (Vaswani et al., 2017) in LLMs holds a significant position in the study of LLM interpretability. The interpretability of attention (Serrano and Smith, 2019; Mrini et al., 2020; Wiegreffe and Pinter, 2019) can generally be summarized in two points: 1. The magnitude of attention weights should correlate positively with the importance of the corresponding positional information; 2. Input units with high weights should have a decisive effect on the output results. Our motivation stems from the first point.

Positional Bias Recent research on positional bias in LLMs has been increasing. In Wang et al. (2023), positional bias is categorised as a part of selection bias. However, its focus is limited to the MCQA (Multiple Choice Question Answering) setting. Another study (Chen et al., 2024) investigates positional bias, or the order sensitivity of models, in mathematical reasoning. It finds that the sequence of rules significantly impacts the final reasoning outcomes, demonstrating that positional bias is an inherent issue in models across multiple tasks. This inspires us to eliminate positional bias from within the model itself.

Debiasing Method Most work attempts to eliminate positional bias from a training perspective. Xiang et al. (2024) tries to eliminate positional bias in in-context learning, where the part of in-context examples is naturally commutative. They introduce a new token-level objective function. Zhang et al. (2024) make the relative position of every token as an external input beside positional embedding to fine-tune LLM.

Wang et al. (2023) tries to mitigate selection bias based on majority vote. The positional bias they mentioned is the model's preference on some positions. They try to estimate the bias distribution and achieve nearly optimal results of majority vote. Li and Gao (2024) tries to eliminate what they called anchored bias by swapping the ground truth label's hidden state with token detected preference. Differences are obvious between our method AAT and these studies. We focus on attention weight and align them at the commutative units' granularity level.

5 Conclusion

This work primarily investigates the issue of positional bias, which is pervasive across various tasks in LLMs. Positional bias causes LLMs to generate significantly different semantic responses to semantically identical inputs arranged in different orders. The problem is formally defined in our study in a general style.

Through extensive empirical analyses, we propose and verify that irregularities in attention weights are one of the primary sources of positional bias. Our proposed debiasing method, AAT, eliminates positional bias by aligning attention weights of specific layers and specific permutations. It achieves excellent results across multiple datasets and models. We emphasize the generality of the AAT method through additional experiments on text, and an analysis of time efficiency also shows that AAT outperforms statistical algorithms, such as majority vote. We hope the empirical analyses in this work and our debiasing method can inspire future research on the bias and robustness of LLMs.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Xinyun Chen, Ryan A Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise order matters in reasoning with large language models. *arXiv preprint arXiv:2402.08939*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. 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)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Ruizhe Li and Yanjun Gao. 2024. Anchored answers: Unravelling positional bias in gpt-2's multiple-choice questions. *Preprint*, arXiv:2405.03205.

Khalil Mrini, Franck Dernoncourt, Quan Hung Tran, Trung Bui, Walter Chang, and Ndapa Nakashole. 2020. Rethinking self-attention: Towards interpretability in neural parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 731–742, Online. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Yong-Min Shin, Siqing Li, Xin Cao, and Won-Yong Shin. 2024. Revisiting attention weights as interpretations of message-passing neural networks. *Preprint*, arXiv:2406.04612.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. *Preprint*, arXiv:1909.11218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Haochun Wang, Sendong Zhao, Zewen Qiang, Nuwa Xi, Bing Qin, and Ting Liu. 2024a. Beyond the answers: Reviewing the rationality of multiple choice question answering for the evaluation of large language models. *Preprint*, arXiv:2402.01349.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *Preprint*, arXiv:2305.17926.

Xinpeng Wang, Chengzhi Hu, Bolei Ma, Paul Röttger, and Barbara Plank. 2024b. Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think. *Preprint*, arXiv:2404.08382.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. 2024. Addressing order sensitivity of in-context demonstration examples in causal language models. *Preprint*, arXiv:2402.15637.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zheng Zhang, Fan Yang, Ziyan Jiang, Zheng Chen, Zhengyang Zhao, Chengyuan Ma, Liang Zhao, and Yang Liu. 2024. Position-aware parameter efficient fine-tuning approach for reducing positional bias in llms. *Preprint*, arXiv:2404.01430.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Limitations

Need extra position of commutative parts. To calculate the average attention weight of one commutative unit in one permutation of commutative part, we need to calculate the token position in this permutation first. This could be different according to different tokenizers.

Different model has different biased layers. Different models exhibit positional bias predominantly in different layers. It is necessary to estimate this based on the specific model in question.

AAT is only verified on Scaled Dot-Product Attention. This paper only validated the effectiveness of the method on the Scaled Dot-Product Attention (Vaswani et al., 2017) and did not verify its effectiveness on other attention mechanisms. There was no comparison of the effects of different attention mechanisms on positional bias.

Only works on open-source LLMs. Our method can only be applied to open-source large models because it operates from within the model, requiring modifications to the model's attention weights.

B Majority Vote

To standardise the model's responses across various problem permutations, the simplest method is to select one answer as the final response. This approach effectively eliminates positional bias, as the answer remains consistent across all tested permutations. The PIR and top-k reach the theoretical upper bound. A strong permutation-based debiasing baseline is shown in Wang et al. (2023); Zheng et al. (2023). It averages the model's prediction distributions under various commutative unit permutations. It's a kind of majority vote method often used to improve the accuracy of models. We refer to this method as a majority vote in the following studies.

C Experiment setup

Both MCQA datasets we used in our experiments are organized in a 4-option format. MMLU encompasses expertise and questions from various fields, while CosmosQA requires the LLM to make selections based on a given passage. All prompts used are shown in Appendix E. Due to the necessity of testing every permutation of each data point, the data volume will be expanded by a factor of 24. Consequently, we selected the first 299 data points from the validation split of CosmosQA and 10 data points from each of the 57 subcategories in the test split of MMLU, amounting to 570 in total. Therefore, the respective volumes of data tested are 7176 and 13680. Except for assessing the disruptive impact, all experiments on MCQA are calculated with the cumulative probability values of the option content (Wang et al., 2024b). All experiments were conducted on 4× A100 GPUs. Except for the 70b model, which was loaded with int8 precision, all models were loaded with bf16 precision.

D Additional Results

We will show the full results and some additional results in this part.

D.1 Difference breakdown on MMLU.

Results of difference on MMLU are shown in Table 5. The result still remains on MMLU.

D.2 Model size results.

We then test AAT on Llama2 with different sizes. 7b, 13b and 70b results are reported in Figure 5.

Results show that AAT's effect causes a disruption in Llama2-70b. The cause behind this can be that we load 70b model in int8 but bf16. Another reasonable explanation is attention weight has less influence on the final output when the dimension of the hidden state gets huge.

D.3 Full results of permutation and layer.

D.4 Full-text results.

D.5 Few-shot learning.

We further test how Llama2-13b-chat performs on the 5-shot setting. Results are shown in Table 4.

AAT works better under few-shot setting according to Table 4.

E Prompts

Shot Num	Method	Prob choice			Text choice			
		Acc	Top-1	PIR	Acc	Top-1	PIR	IR↓
0-shot	origin	0.5842	0.8119	0.3144	0.5812	0.7347	0.1940	0.0619
	AAT	0.5928	0.9691	0.8729	0.5548	0.8611	0.5953	0.1176
	AAT EF	0.5994	0.9472	0.7860	0.6324	0.8712	0.5619	0.0683
5-shot	origin	0.6731	0.8434	0.4348	0.6678	0.8297	0.3712	0.0188
	AAT	0.6970	0.9797	0.9097	0.6858	0.9508	0.8796	0.0438
	AAT EF	0.6998	0.9567	0.7993	0.7111	0.9561	0.8428	0.0116

Table 4: AAT's performance on few-shot; tested on CosmosQA

Model	Method	PM	T->F% ↓	F->F(%)	F->T(%)↑	Avg Diff(%)
Llama2	MV	1	36.9	33.1	30.0	30.7
	AAT	1	**34.6**	33.5	**31.91**	30.04
	MV	4	28.6	35.0	36.5	23.9
	AAT	4	**27.7**	34.0	**38.3**	25.2
	MV	24	**24.4**	33.1	**42.5**	23.1
	AAT	24	26.2	33.0	40.8	24.3
ChatGLM3	MV	1	34.6	36.3	29.1	31.2
	AAT	1	**32.3**	37.6	**30.1**	31.2
	MV	4	30.6	36.6	32.8	24.7
	AAT	4	**28.3**	36.2	**35.5**	26.7
	MV	24	26.5	36.5	37.0	23.5
	AAT	24	**24.9**	34.6	**40.5**	25.6
Qwen1.5	MV	1	**37.1**	26.9	**35.9**	22.0
	AAT	1	38.3	27.7	34.0	22.6
	MV	4	**28.2**	25.3	**46.5**	18.8
	AAT	4	28.4	25.9	45.7	19.4
	MV	24	**26.9**	25.1	**48.0**	17.9
	AAT	24	29.9	25.4	44.7	18.9

Table 5: The breakdown results of models' difference from origin on MMLU. PM=1 for single permutation, 4 for cyclic permutation and 24 for full permutation.

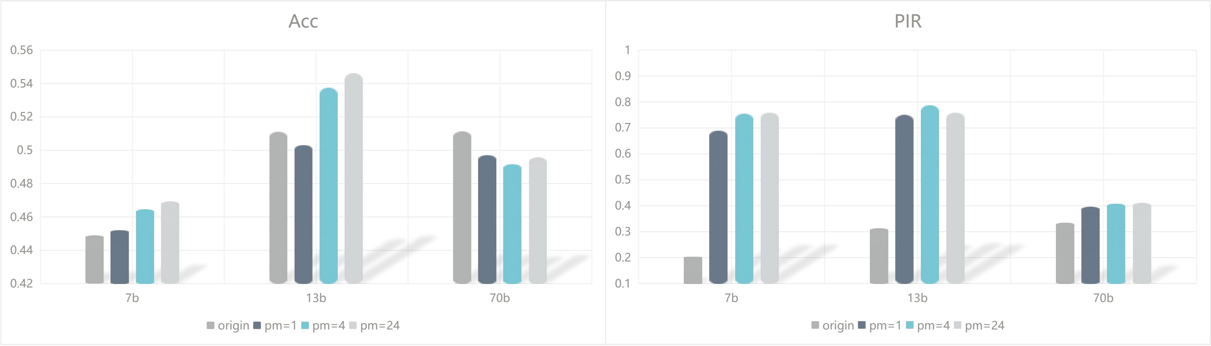


Figure 5: Acc and PIR of different size models on MMLU; 3 permutations of AAT are reported.

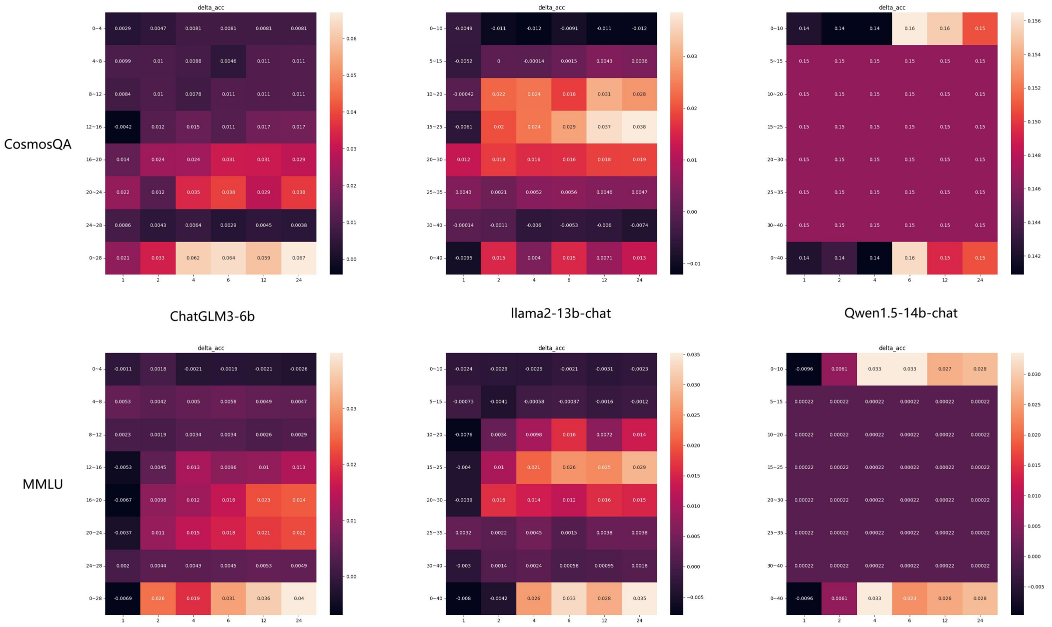


Figure 6: Heatmap of ΔAcc ($\Delta Acc = Acc(debiased) - Acc(origin)$) on different layer and permutation set; the last line indicates the full layer.

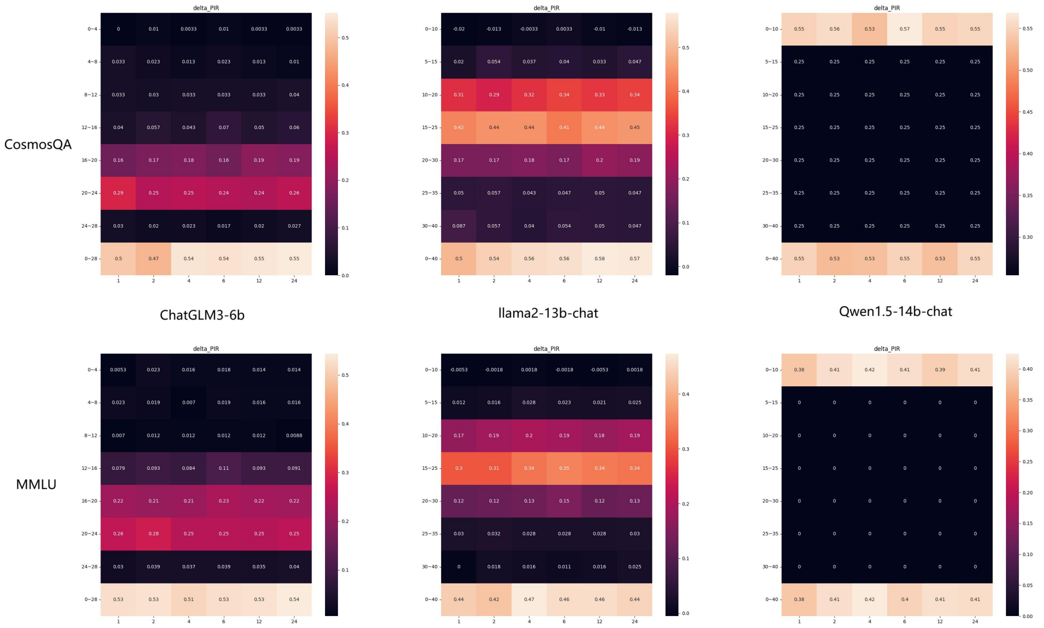


Figure 7: Heatmap of ΔPIR ($\Delta PIR = PIR(debiased) - PIR(origin)$) on different layer and permutation set; the last line indicates the full layer.

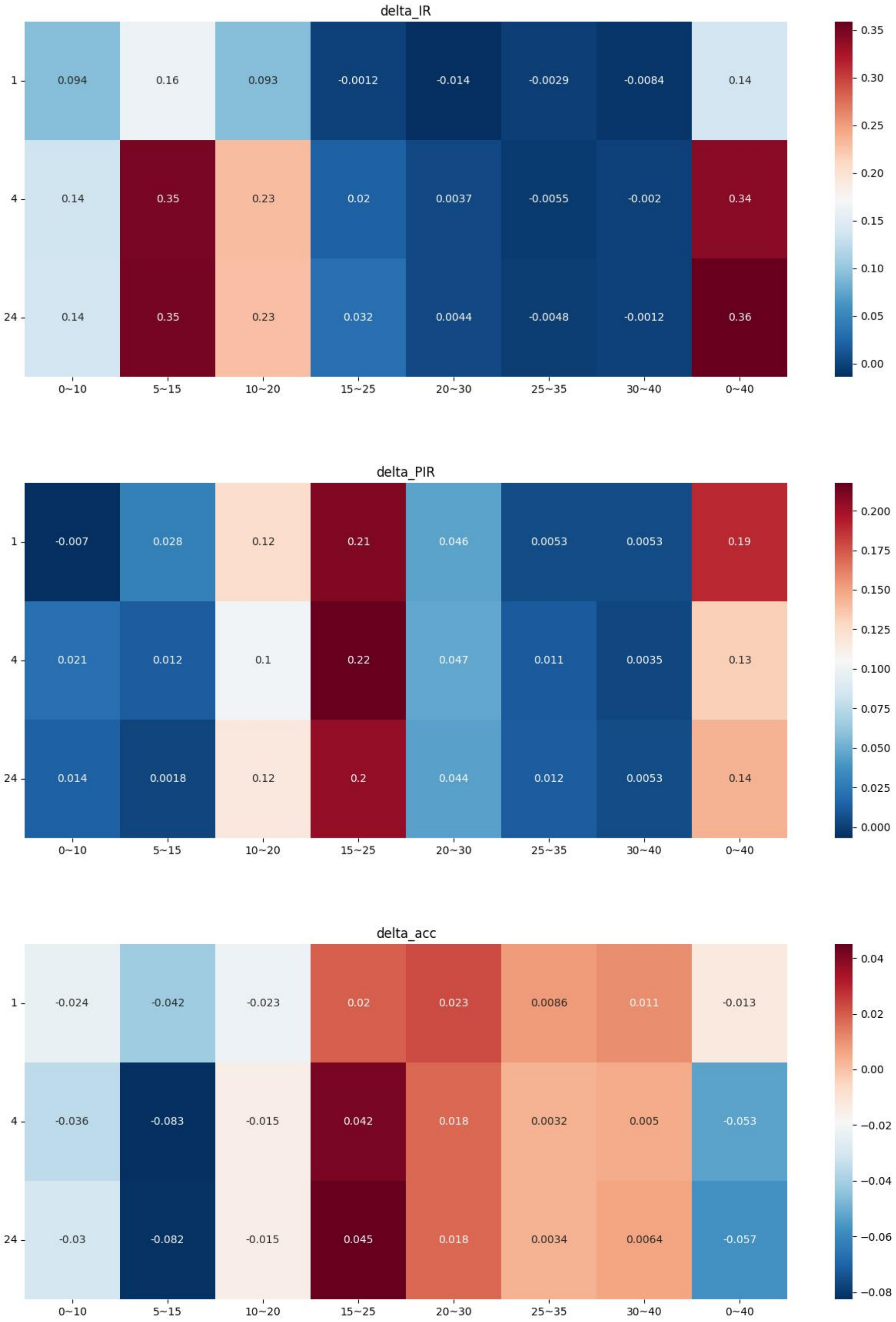


Figure 8: ΔIR, PIR, Acc of Llama2-13b-chat heatmaps on MMLU;

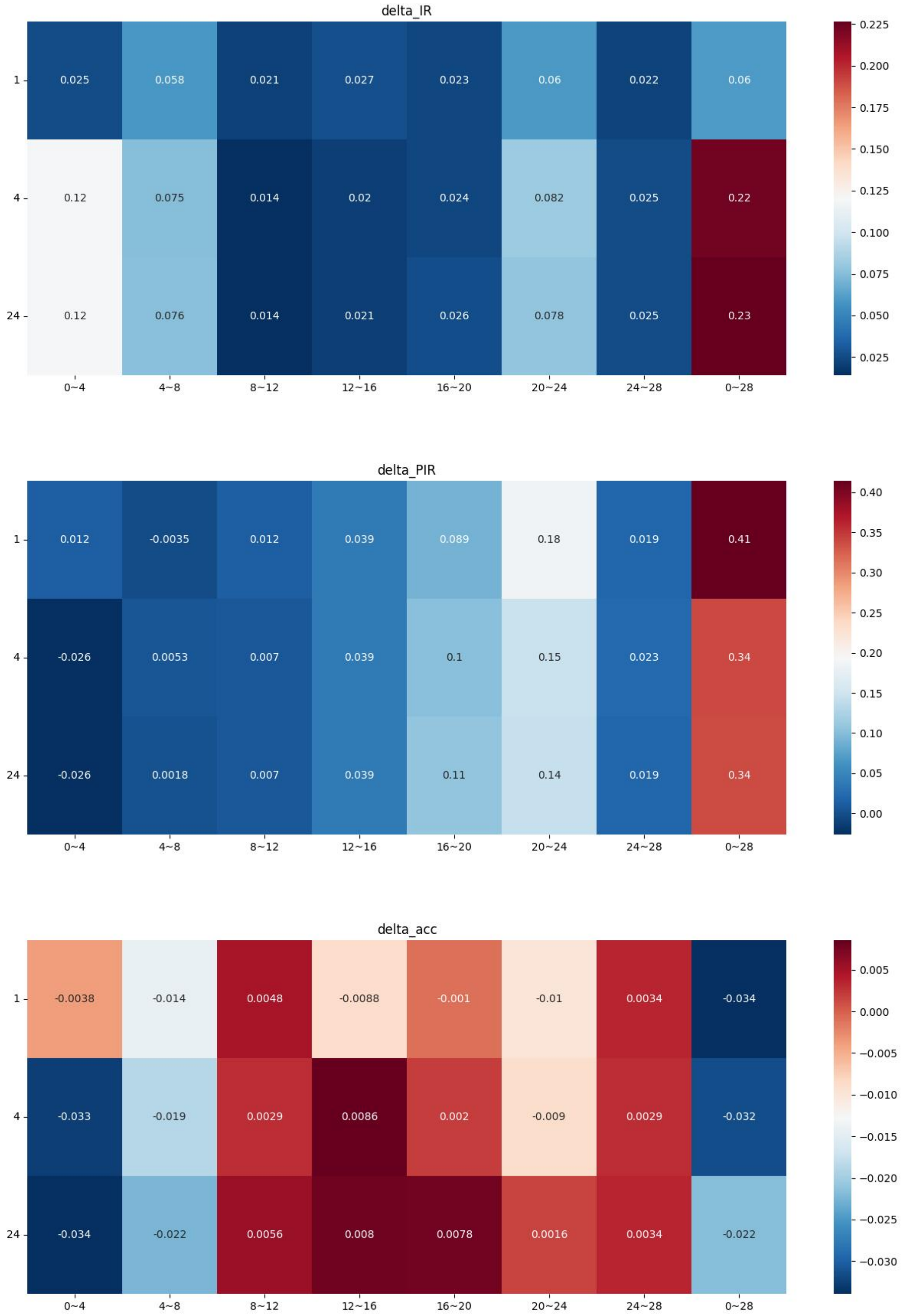


Figure 9: ΔIR, PIR, Acc of ChatGLM3-6b heatmaps on MMLU;

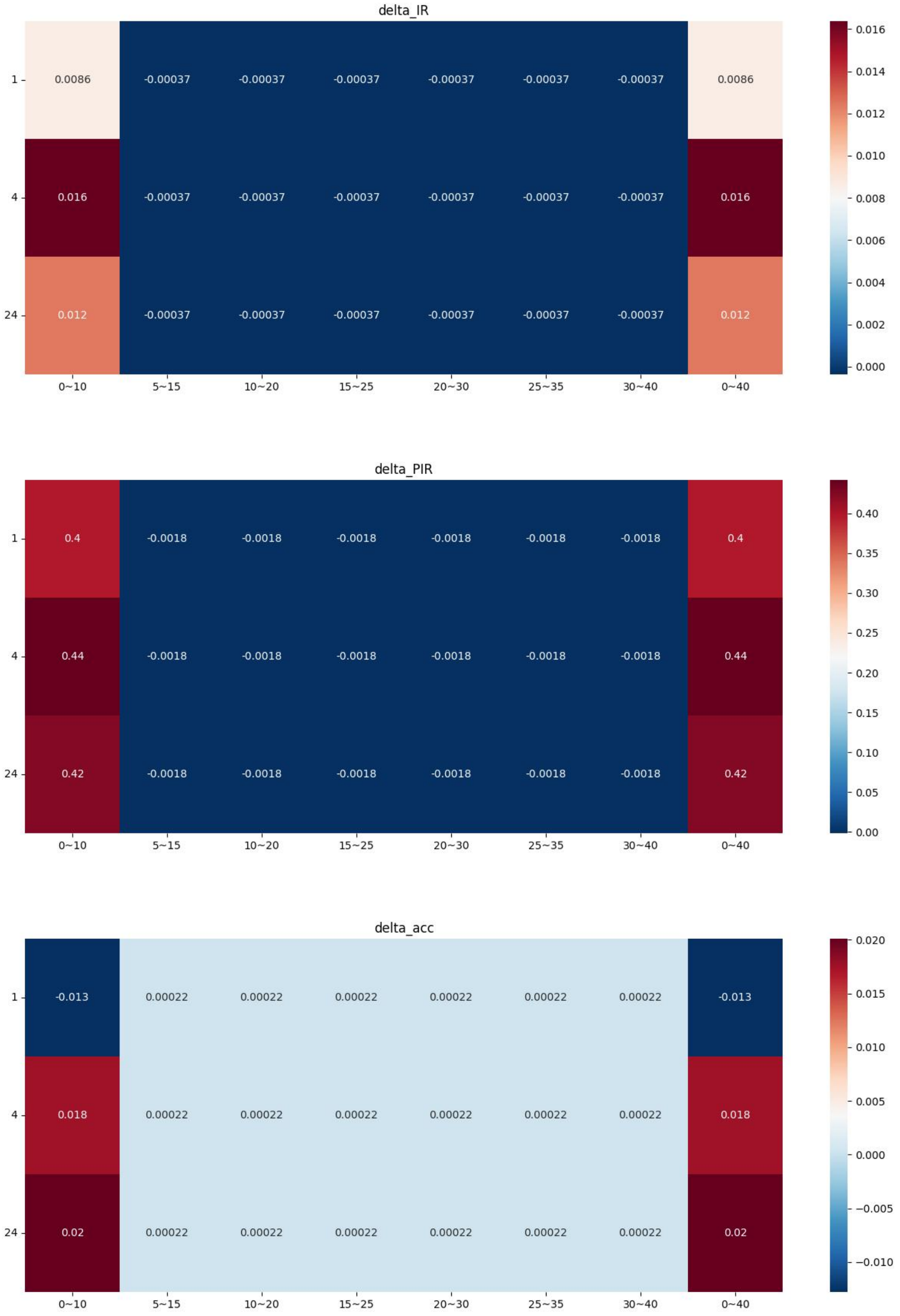
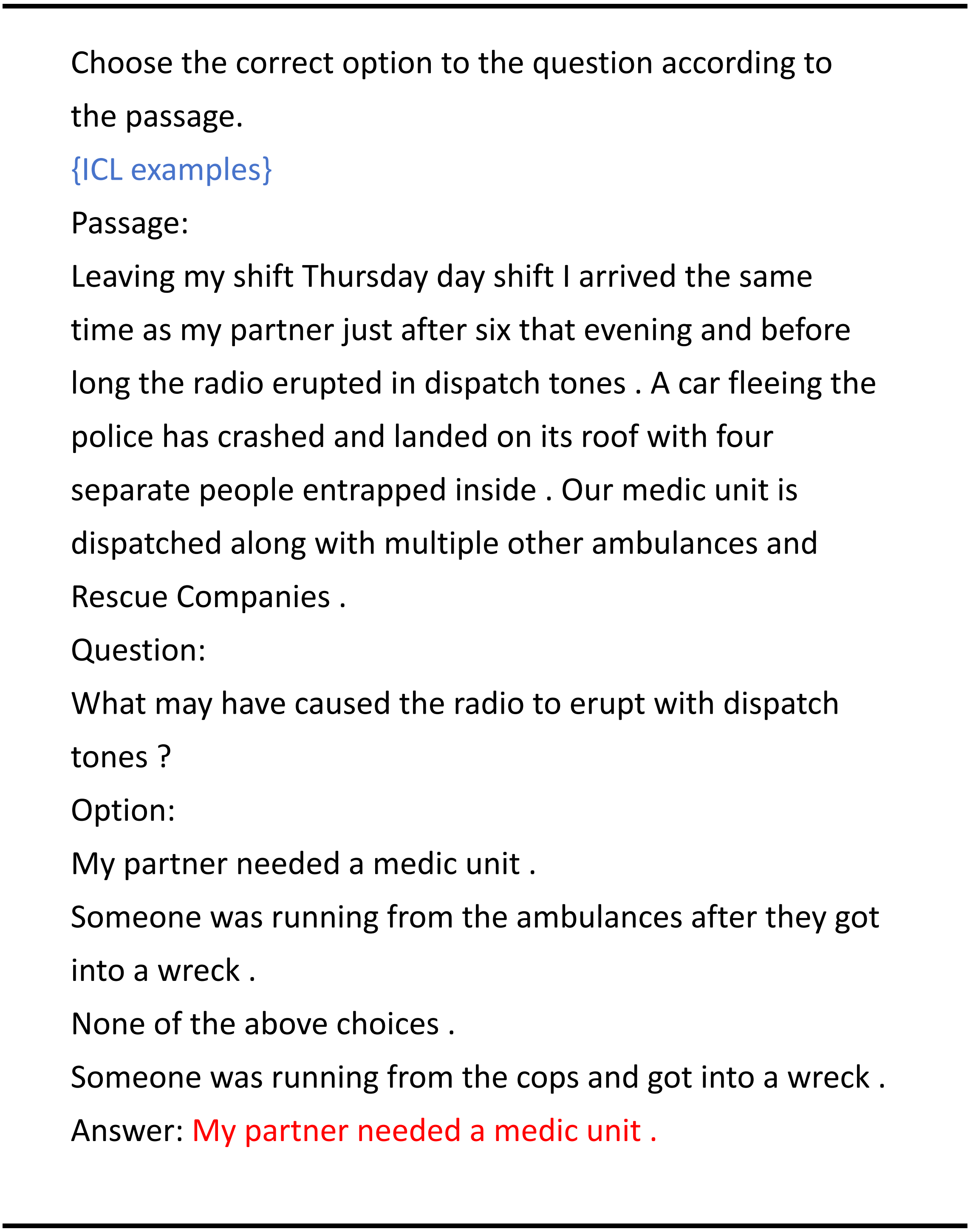


Figure 10: ΔIR, PIR, Acc of Qwen1.5-14b-chat heatmaps on MMLU;



Choose the correct option to the question according to the passage.

{ICL examples}

Passage:

Leaving my shift Thursday day shift I arrived the same time as my partner just after six that evening and before long the radio erupted in dispatch tones . A car fleeing the police has crashed and landed on its roof with four separate people entrapped inside . Our medic unit is dispatched along with multiple other ambulances and Rescue Companies .

Question:

What may have caused the radio to erupt with dispatch tones ?

Option:

My partner needed a medic unit .

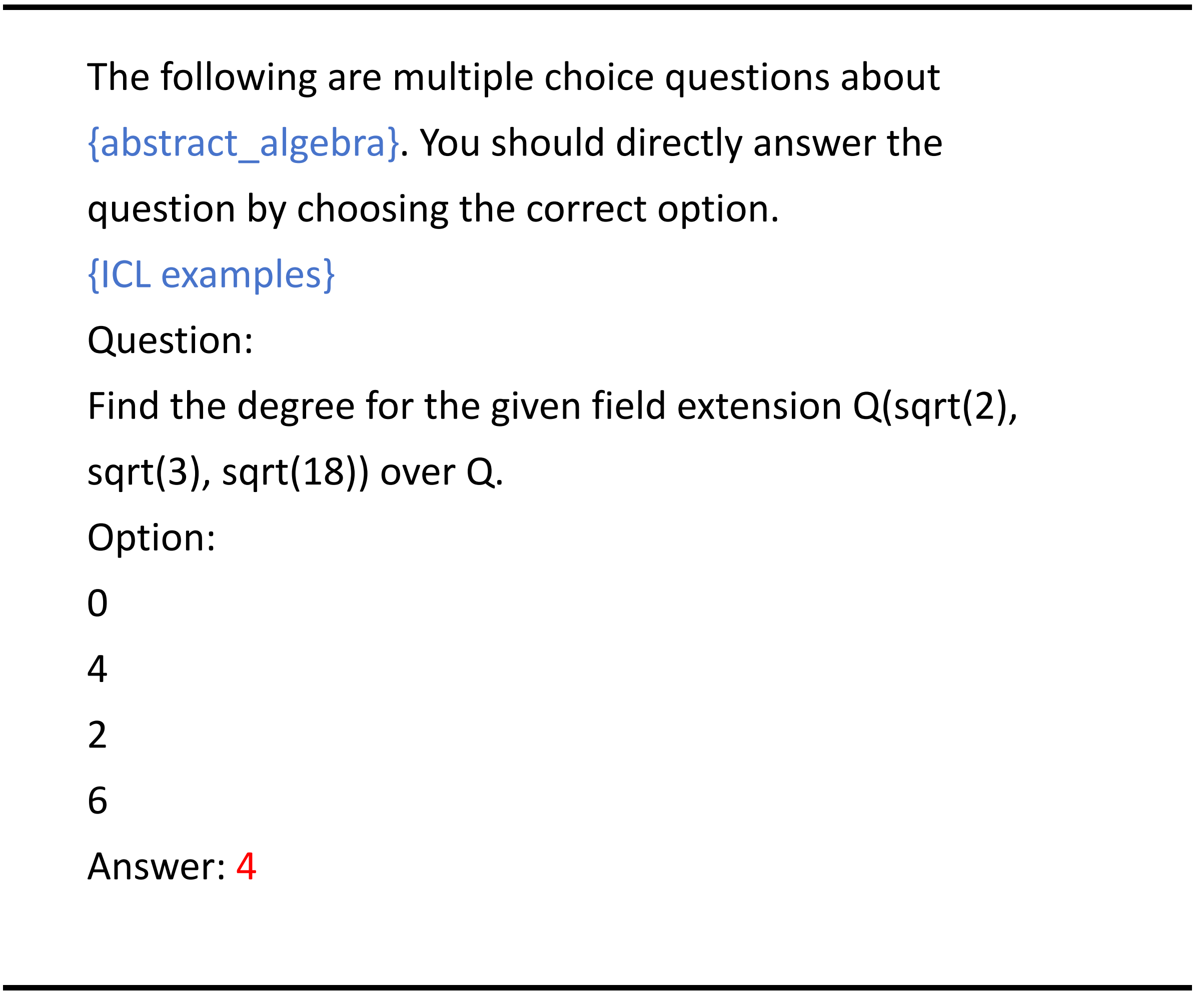
Someone was running from the ambulances after they got into a wreck .

None of the above choices .

Someone was running from the cops and got into a wreck .

Answer: My partner needed a medic unit .

Figure 11: Prompt of CosmosQA used in the paper.



The following are multiple choice questions about {abstract_algebra}. You should directly answer the question by choosing the correct option.

{ICL examples}

Question:

Find the degree for the given field extension Q(sqrt(2), sqrt(3), sqrt(18)) over Q.

Option:

0

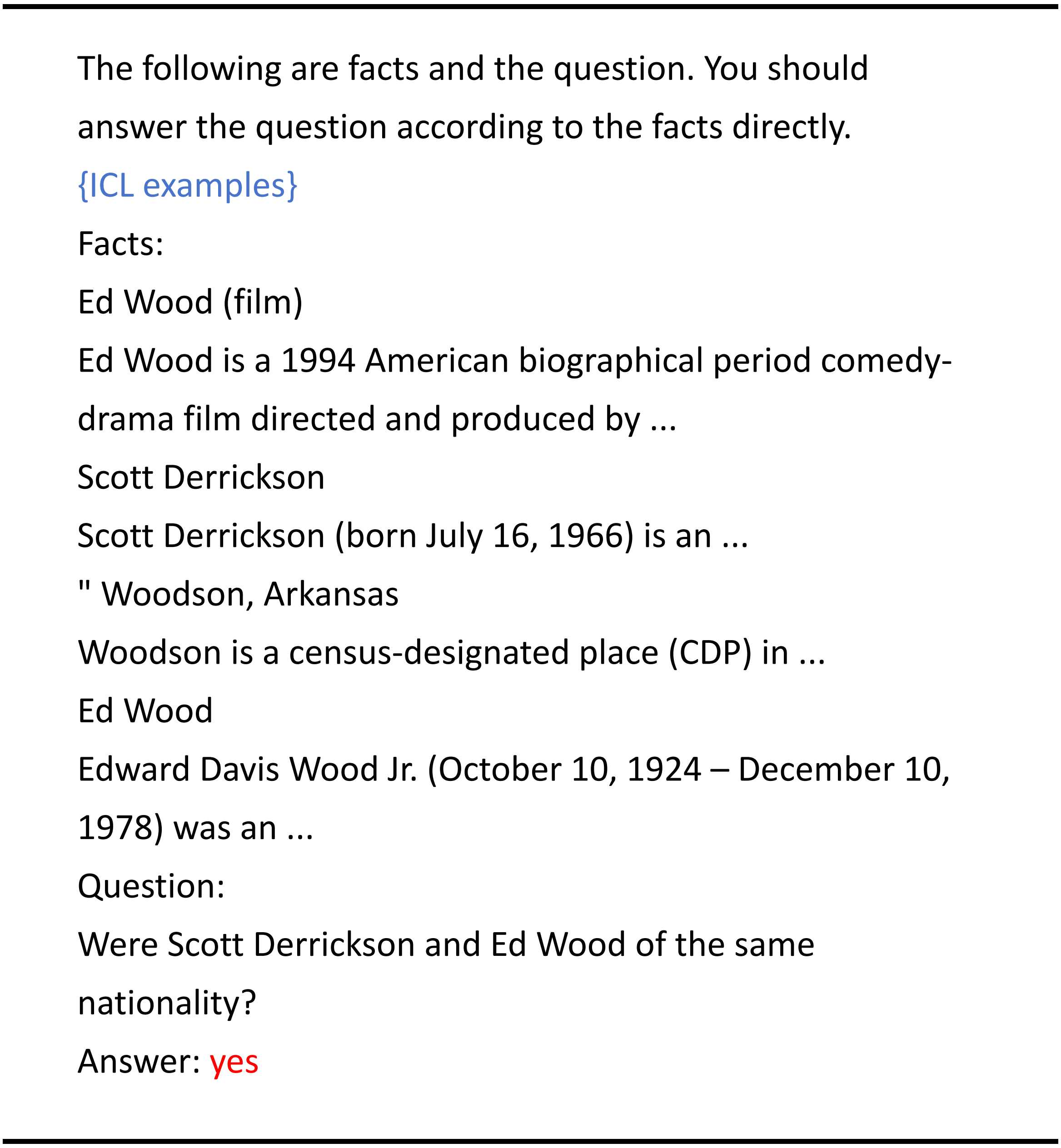
4

2

6

Answer: 4

Figure 12: Prompt of MMLU used in the paper.



The following are facts and the question. You should answer the question according to the facts directly.

{ICL examples}

Facts:

Ed Wood (film)

Ed Wood is a 1994 American biographical period comedy-drama film directed and produced by ...

Scott Derrickson

Scott Derrickson (born July 16, 1966) is an ...

" Woodson, Arkansas

Woodson is a census-designated place (CDP) in ...

Ed Wood

Edward Davis Wood Jr. (October 10, 1924 – December 10, 1978) was an ...

Question:

Were Scott Derrickson and Ed Wood of the same nationality?

Answer: yes

Figure 13: Prompt of HotpotQA used in the paper.