Guided Permutation Scoring (GPS) : Option-Order Permutation for Multiple-Choice Question Answering

Anonymous EMNLP submission

Abstract

Large Language Models (LLMs) are sensitive to the order in which answer choices are presented in Multiple-Choice Question Answering. The same question can yield a different prediction when its options are shuffled. Although exhaustive majority voting over all permutations can mitigate this bias, its factorial cost makes it impractical at inference time. We propose **GPS** (Guided Permutation Scoring), a two-stage inference-time pipeline that selects option orderings more likely to produce correct predictions from a frozen LLM. A sensitivity classifier first identifies which questions are genuinely vulnerable to positional interference; robust questions are answered directly, while sensitive ones are forwarded to a trained permutation scorer that identifies a more position-stable option ordering, promoting better predictions. Unlike prior approaches that either enumerate all permutations uniformly, or require modifying the LLM's parameters, GPS is the first method to actively select permutations so that the base model can consistently give better answers. GPS steadily improves accuracy and Macro F1 across all evaluated model-dataset combinations, with remarkable reductions in positional bias metrics.

Code Availability The code used to run the experiments is available in the GitHub repository [1].

1 Introduction

Multiple-Choice Question Answering (MCQA) is one of the most widely used frameworks for evaluating LLMs, yet it exposes a basic reliability problem: models are influenced not only by the semantic content of answer choices, but also by their positions. When the same choices are shuffled, an LLM may change its prediction, a phenomenon known as option-order bias. The average accuracy gap between single-pass inference and exhaustive Majority Voting (MV) over all possible permutations of option-choices is substantial (see Figure 1) : 5.0 percentage points on ARC Challenge (Clark et al., 2018), 4.0 points on MMLU-Redux-2.0 (Gema et al., 2025), and 8.7 points on CommonsenseQA (Talmor et al., 2019). These gaps indicate that standard evaluation leaves recoverable accuracy unrealized.

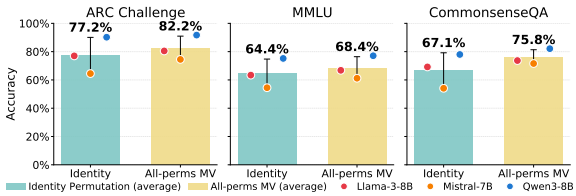

Figure 1: Original (identity) order MCQA vs. exhaustive Majority Voting (MV) avg. accuracy across models and datasets under zero-shot prompting. Positional bias causes the original inference to underperform, favoring certain answer slots, a bias MV effectively fixes.

While MV provably eliminates positional bias, it scales factorially (4!=24, 5!=120), making it prohibitively expensive. Random subsampling offers no guarantee of picking informative orderings. Calibration methods (Zheng et al., 2024; Li et al., 2025) adjust output probabilities post-hoc, but do not enforce consistency across permutations. Training-time approaches such as LoRA fine-tuning (Guda et al., 2025) and permutation-aware alignment (Zheng et al., 2026) are effective, but require updating model parameters, a major constraint when the base LLM must remain frozen.

This raises a natural question. *Can we reduce this option-order bias efficiently, without modifying the LLM, and without running all possible permutations?* Our key observation is that not all questions are equally vulnerable. Many are answered correctly regardless of option order, and we refer to these as robust questions. Only a targeted subset

[1] https://anonymous.4open.science/r/gps-73EF/

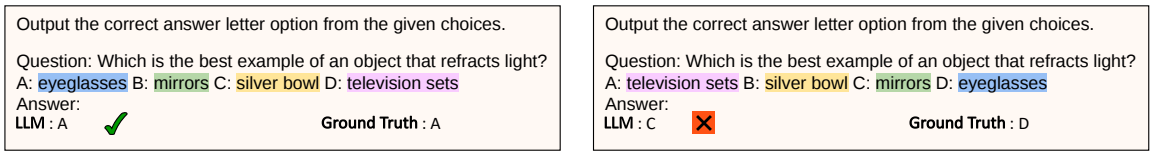

Figure 2: Examples of the identity permutation π_1 (left), and the reverse permutation $\pi_{N!}$ (right) applied to the same question. The two passes present answer choices in opposite orders. The LLM response is not invariant to the order.

of questions genuinely requires intervention. If we could identify order-sensitive questions, and intelligently select a better-suited option-permutation for them, we could improve over direct single-pass answering while closing the gap toward MV.

Hence, we propose **GPS** (Guided Permutation Scoring), a two-stage inference-time pipeline built entirely on top of an adaptation-free LLM. A lightweight sensitivity classifier processes two diagnostic LLM passes per question, one with the options in their original order, one with the options fully reversed (see Figure 2), and extracts scalar confidence features from both to classify each question as either positionally sensitive or robust. These features provide a low-cost probe of positional sensitivity by exposing the model to maximally different option placements. Robust questions are answered directly from the original-order pass, at no additional cost. Sensitive questions proceed to a learned permutation scorer that selects top-k high-quality permutations (k=1 in our main experiments) for a dedicated LLM inference. GPS consistently improves accuracy and Macro F1 across a variety of model-dataset combinations, with significant bias reductions. A comparison with a random permutation baseline confirms that GPS gains reflect genuine learned signal rather than chance reorderings. GPS generalizes across a wide range of model architectures, datasets, and prompting strategies like In-Context Learning (**ICL**) (Dong et al., 2024), and Chain-of-Thought (**CoT**) (Wei et al., 2023). Our key contributions are:

- To the best of our knowledge, GPS is the first method that *corrects* the permutation of options rather than running them uniformly.
- GPS introduces a parameter-efficient permutation scorer that consistently surfaces high-quality orderings from the model's own hidden-state representations, generalizing across diverse models and datasets.
- GPS operates as a fully modular, plug-and-play component on top of any frozen LLM, requiring neither weight modification nor architectural adaptation.

2 Related Work on Bias Mitigation Strategies

Selection bias in LLM-based Multiple-Choice Question Answering has been widely observed, and systematically quantified in prior work (Robinson et al., 2023; Wang et al., 2023). These studies show that model predictions can be affected by non-semantic factors such as option order, answer labels, and token choices (Pezeshkpour and Hruschka, 2024; Wei et al., 2024; Reif and Schwartz, 2024). Existing mitigation methods can be broadly grouped into inference-time correction and training-time debiasing approaches.

Inference-time Correction. Inference-time correction methods avoid updating model parameters, and instead modify prompts, calibrate model outputs, or aggregate predictions over option permutations. For example, prompt-based Auxiliary Option Injection (AOI) (Choi et al., 2025) introduces an additional abstention option to reduce forced selection bias, but its effectiveness is often prompt-sensitive. Calibration-based methods adjust model outputs to correct biased option preferences. PriDe (Zheng et al., 2024) estimates option-identifier priors from a small set of permuted examples for normalization, while CalibraEval (Li et al., 2025) learns a projection function to calibrate skewed prediction distributions. Recent work also studies label-bias calibration to reduce systematic preferences over output labels (Reif and Schwartz, 2024). However, calibration-based methods do not directly enforce consistency across option permutations.

Permutation-based aggregation addresses option-order sensitivity by evaluating multiple option permutations and aggregating predictions. While majority voting improves robustness, exhaustive enumeration scales factorially with the number of options; Batch Question-Context KV caching (Guda et al., 2025) improves its efficiency, but still applies

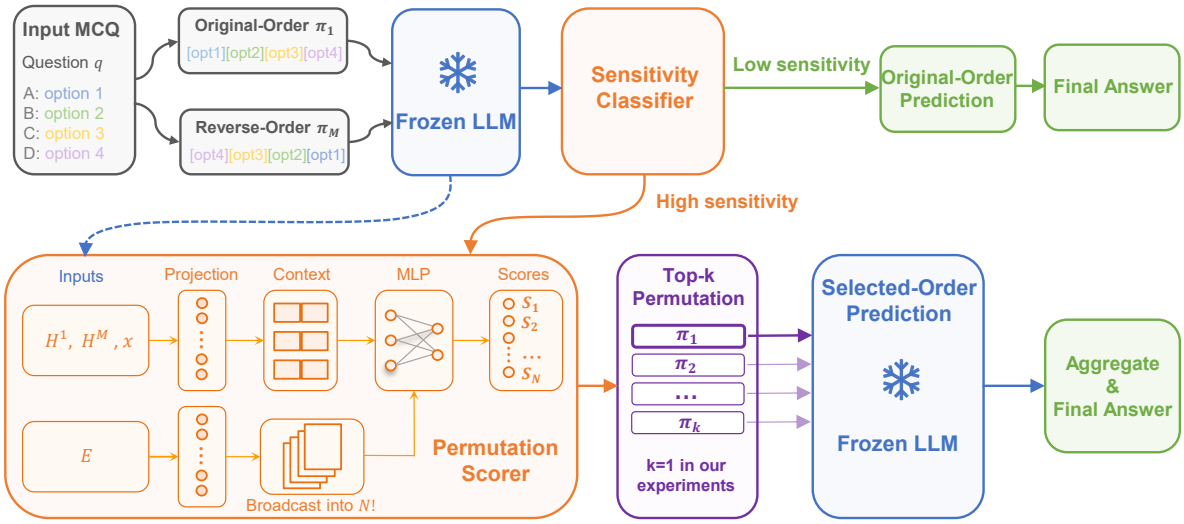

Figure 3: Overview of the GPS pipeline. *Top row :* Each input MCQ is processed under, original-order (π_1), and reversed (π_M), by a frozen LLM, which extracts embeddings, and scalar features ($E, H^1, H^{N!}, \Delta, p_{\max}$, agreement, difference). A sensitivity classifier routes High-sensitivity questions to the Permutation Scorer. *Bottom row :* The scorer takes ($H^1, H^{N!}, E$), and scalar features (x) as input, and scores all candidate orderings in a single batched operation, and selects the top-k permutations (k=1 in our experiments) for a dedicated frozen LLM inference pass.

permutation-based inference uniformly to all questions. In contrast, GPS makes permutation-based correction selective, using compact auxiliary networks to identify sensitive questions, and select a small subset of candidate permutations, reducing unnecessary inference calls.

Training-time Debiasing. Some methods go beyond output-level correction by intervening in model internals. UniBias (Zhou et al., 2024) and Bias Node Pruning (Choi et al., 2025) identify and remove internal sources of biased selection behavior. Training-time debiasing methods further update model behavior through additional optimization, including task-independent LoRA adapters (Guda et al., 2025), teacher–student distillation (Liusie et al., 2024), multiple-choice symbol-binding enhancement (Xue et al., 2024), and permutation-aware alignment such as PA-GRPO (Zheng et al., 2026). Although training-time methods can reduce selection bias, updating the LLM itself is often costly. Our approach instead trains only low-overhead auxiliary networks while keeping the base LLM frozen.

3 Problem Definition

Let q be a multiple-choice question with N answer options $\mathcal{O} = \{o_1, o_2, ..., o_N\}$ and ground-truth label $y^* \in \{\text{A}, \text{B}, \text{C}, ...\}$. A permutation $\pi \in \mathcal{S}_N$ reorders the options so that position j receives option $o_{\pi(j)}$; the identity permutation $\pi_1 = (0, 1, ..., N-1)$ corresponds to the original presentation order. We write $f_\theta(\pi; q)$ for the next-token logit-argmax prediction of a frozen language model f_θ on the prompt built with permutation π. We define the *identity-order prediction* as $\hat{y}_1 = f_\theta(\pi_1; q)$.

A model exhibits option-order bias when $f_\theta(\pi; q)$ varies across permutations even though the correct answer is invariant. Our goal is to detect which test samples are affected by this bias, and correct for it *without* running all $|\mathcal{S}_N|$=$N!$ permutations per sample at test time. We use the constant $M = N!$ in the remaining part of this paper. We denote $\pi_M = (N-1, ..., 1, 0)$ to be the reverse order of option-choices. Below is the *prompt structure* used in all our experiments. For CommonsenseQA dataset (five options), the suffix extends to option E.

```
prompt = f"""Output the correct answer letter
↪  option from the given choices.
Question: {question}
A: {opt[1]} B: {opt[2]} C: {opt[3]} D: {opt[4]}
Answer:"""
```

4 Method

GPS works in two stages: (1) *sensitivity classification*, and (2) *permutation scoring*, (see Figure 3).

The sensitivity classifier assigns a binary label (either 'low' or 'high') to each sample. Low-sensitivity samples directly use the single-pass pre-

diction result. High-sensitivity samples proceed to the final stage, where the permutation scorer selects a small top-k subset of permutations, only these k passes are evaluated, and the resulting predictions are aggregated by majority vote. We use $k = 1$ in our main experiments.

4.1 Sensitivity Classifier

The sensitivity classifier is a lightweight logistic regressor, adding negligible overhead, and requiring no LLM fine-tuning. At First, each question prompt undergoes two LLM passes (see Figure 2), with (1) the identity order π_1, and (2) the reverse order π_M of option-choices. We extract scalar features from these two inferences for the sensitivity classifier. They are :
1) output-entropy $H(\pi) = -\sum_j p_j \log p_j$,
2) max-probability $p_{(1)}$,
3) confidence margin $\Delta(\pi) = p_{(1)} - p_{(2)}$ (gap between top-two softmax probabilities).
We get six scalar features from the two passes. Four cross-pass features are then appended : the pass-agreement indicator $\mathbf{1}[\hat{y}_1{=}\hat{y}_M]$, entropy difference $H^1{-}H^M$, margin difference $\Delta^1{-}\Delta^M$, and minimum max-probability $\min(p^1_{(1)}, p^M_{(1)})$. Finally, we build a feature vector $\mathbf{x} \in R^{10}$ (see Appendix A).

$$\mathbf{x} = \big[H^1,\ \Delta^1,\ p^1_{(1)},\ H^M,\ \Delta^M,\ p^M_{(1)},\ \mathbf{1}[\hat{y}_1{=}\hat{y}_M],\ H^1{-}H^M,\ \Delta^1{-}\Delta^M,\ \min(p^1_{(1)}, p^M_{(1)})\big],$$

where p_j is the probability of the j-th option letter, $p_{(k)}$ is the k-th highest probability among all N option probabilities, and superscripts 1 and M refer to π_1 and π_M respectively.

We train the classifier on the training-set features $\{\mathbf{x}^{(n)}, s^{(n)}\}$ where the binary label is decided based on the median permutation sensitivity ($\tilde{\sigma}$) over the training set.

$$s^{(n)} = \mathbf{1}\left[\sigma_{q^{(n)}} \geq \tilde{\sigma}\right], \tag{1}$$

Permutation sensitivity of a sample is defined as

$$\sigma_q = \mathrm{std}\big(\{p_{\text{correct}}(\pi_i)\}_{i=1}^{M}\big), \tag{2}$$

where p_{correct} is the probability of the correct option letter among all permutations of the option texts (see Appendix C). At test time, *low-sensitivity* samples ($\hat{s}{=}0$) are answered directly via $\hat{y}_1$, while *high-sensitivity* samples ($\hat{s}{=}1$) are forwarded to the permutation scorer. The sensitivity classifier routes 37% to 52% of test samples to the top-k GPS path (see Figure 7, Appendix B).

4.2 Permutation Scorer

The Permutation Scorer g_ϕ is a small neural network trained to predict which option ordering is most likely to elicit a correct response from the frozen LLM. We extract three sets of permutation-aware option representations from the previous two LLM inferences (π_1, π_M).

1. **Token Embeddings** $E \in R^{N\times d}$: mean-pooled embeddings of each option text, taken from the token embedding layer of the LLM.
2. **Original order hidden states** $H^{(1)} \in R^{N\times d}$: mean-pooled last-layer hidden states of each option text, from the identity-order LLM pass π_1.
3. **Reverse order hidden states** $H^{(M)} \in R^{N\times d}$: same as $H^{(1)}$ but from the reverse-order pass π_M. Together with $H^{(1)}$, they give a two-view signal of positional bias.

These representations capture complementary signals : token embeddings encode order-independent option semantics, while original and reverse hidden states reflect order-dependent contextual changes. Each sample is also associated with the scalar feature vector $\mathbf{x}$ from the previous stage. A *correctness label vector* $\mathbf{c} \in \{0,1\}^M$ is computed for all the training samples with binary values for each permutation of option orders, where $c_i = \mathbf{1}[f_\theta(\pi_i; q) = y^*]$, for $i = 1, \ldots, M$. We train the *PermScorer* g_ϕ to predict $\mathbf{c}$ from $(E, H^{(1)}, H^{(M)}, \mathbf{x})$. g_ϕ is invoked only for high-sensitivity samples at inference time.

Architecture. Let d be the LLM hidden dimension, p a projection dimension, and h a hidden dimension. The scorer operates as follows:

$$\mathbf{e}_j = W_{\text{opt}}\, E_j \in R^p, \quad j = 1, \ldots, N \qquad \text{(Shared option projection)}$$

$$\mathbf{c}_1 = \tfrac{1}{N}\textstyle\sum_j W_{\text{opt}}\, H^{(1)}_j, \quad \mathbf{c}_M = \tfrac{1}{N}\textstyle\sum_j W_{\text{opt}}\, H^{(M)}_j \qquad \text{(Context Pooling)}$$

$$\mathbf{ctx} = \mathrm{ReLU}(\mathrm{LN}(W_{\text{ctx}}\,[\mathbf{c}_1; \mathbf{c}_M; \mathbf{x}])) \in R^h \qquad \text{(Sample context)}$$

$$\mathbf{v}_i = \big[\mathbf{e}_{\pi_i(1)}; \ldots; \mathbf{e}_{\pi_i(N)}; \mathbf{ctx}\big], \quad i = 1, \ldots, M$$

$$s_i = \mathrm{MLP}(\mathbf{v}_i) \in R. \qquad \text{(Permutation Scoring)}$$

All $M = N!$ scores are computed in a single batched matrix operation by gathering $\mathbf{e}$ according to the (pre-fixed) permutation index tensor $\Pi \in Z^{M\times N}$. Additionally, $W_{\text{opt}} \in R^{d\times p}, W_{\text{ctx}} \in R^{2p+10\times h}$. We have $p = 64$ and $h = 256$.

Metric		MMLU-Redux 2.0 (K=1)			CommonsenseQA (K=1)			ARC Challenge (K=1)		
		Llama-3-8b	Mistral-7b	Qwen3-8b	Llama-3-8b	Mistral-7b	Qwen3-8b	Llama-3-8b	Mistral-7b	Qwen3-8b
Acc. [↑]	Baseline	63.36%	54.52%	75.22%	69.04%	53.73%	77.89%	76.96%	64.51%	90.19%
	Random	63.01%	53.58%	74.34%	69.29%	53.15%	76.19%	77.33%	63.65%	90.47%
	GPS	**64.39%**	**57.04%**	**75.63%**	**70.68%**	**59.95%**	**78.21%**	**78.24%**	**66.89%**	**91.38%**
Macro F1 [↑]	Baseline	63.33%	53.97%	75.13%	69.04%	51.83%	77.71%	76.94%	64.30%	90.28%
	Random	62.95%	53.51%	74.25%	69.26%	53.14%	76.16%	77.28%	63.58%	90.53%
	GPS	**64.31%**	**56.97%**	**75.55%**	**70.68%**	**59.36%**	**78.20%**	**78.21%**	**66.88%**	**91.41%**
RSD [↓]	Baseline	6.10%	20.32%	1.80%	5.67%	38.90%	11.22%	3.39%	18.74%	2.24%
	Random	3.97%	20.00%	**1.55%**	6.38%	40.51%	11.67%	2.63%	21.10%	**2.16%**
	GPS	**2.03%**	**5.19%**	1.76%	**2.24%**	**22.21%**	**1.49%**	**1.65%**	**9.78%**	2.26%
CKLD [↓]	Baseline	1.28%	4.54%	0.07%	0.67%	19.48%	1.46%	0.42%	4.84%	**0.02%**
	Random	**0.23%**	3.99%	0.05%	0.56%	19.15%	2.07%	0.13%	5.52%	0.09%
	GPS	0.30%	**0.18%**	**0.02%**	**0.10%**	**3.95%**	**0.01%**	**0.02%**	**0.72%**	0.05%

Table 1: Results across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge ($k = 1$) for Baseline (original order), Random (uniformly sampled permutation), and GPS. **Bold** = best value per cell; underline = second best. [↑] higher is better; [↓] lower is better. GPS reduces Mistral-7B's CKLD in the most dramatic way on CommonsenseQA.

Training objective. g_ϕ is trained with weighted binary cross-entropy:

$$\mathcal{L}(\phi) = -\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{M}\Big[\alpha\, c_i^{(j)} \log \sigma(s_i^{(j)}) + (1 - c_i^{(j)}) \log\Big(1 - \sigma(s_i^{(j)})\Big)\Big], \quad (3)$$

where σ is the sigmoid function and $\alpha > 1$ is a positive class weight that compensates for the imbalance between correct and incorrect permutations. At test time, the final stage replaces the exhaustive M-pass majority vote with a k-pass budget selected by the learned scorer.

Prediction Estimation. Let $\mathbf{z} \in R^{|V|}$ be the output logit vector of the first generated token across the vocabulary V. For each choice (e.g., `'A'`), logits are aggregated over surface forms $z[\text{`A'}]+z[\text{`_A'}]$, and softmax is applied over all N aggregated values (Choi et al., 2025).

$$P(o_i \mid \mathbf{z}) = \frac{\exp(z[o_i])}{\sum_{k=1}^{N}\exp(z[o_k])}, \quad i = 1, \ldots, N. \quad (4)$$

The total LLM cost for a high-sensitivity sample is $2 + k$ passes (details in Appendix D).

5 Experiments

We evaluate GPS on open-weight LLMs : Llama-3-8B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), Qwen3-8B (Team, 2025). We use three Multiple-Choice Question Answering benchmarks: MMLU-Redux-2.0 (Gema et al., 2025), ARC Challenge (Clark et al., 2018), and CommonsenseQA (Talmor et al., 2019) to assess GPS (see Appendix E).

5.1 Evaluation Metrics

Following (Choi et al., 2025), we evaluate GPS on four metrics to examine both predictive accuracy, and positional bias in model responses. The metrics are : Accuracy, Macro F1, Relative Standard Deviation (RSD), and Choice Kullback-Leibler Divergence (CKLD). **Accuracy** is the ratio of questions answered correctly. **Macro F1** computes the F1 score per answer slot, and averages them equally, penalizing models that are biased toward specific choice positions. **RSD** measures input-position bias by quantifying how much per-slot accuracy varies around its mean:

$$\text{RSD} = \frac{1}{\bar{s}}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2} \quad (5)$$

where s_i is the accuracy when the correct answer appears in slot i, and $\bar{s}$ is the mean slot accuracy. **CKLD** measures output-distribution bias by comparing the empirical distribution of gold labels to that of predicted labels:

$$\text{CKLD} = \sum_{i=1}^{N} P_{\text{gold}}(i) \log \frac{P_{\text{gold}}(i)}{P_{\text{pred}}(i)} \quad (6)$$

Lower RSD indicates less sensitivity to answer ordering while lower CKLD indicates better calibration of predicted to true label distributions.

5.2 Results

Table 1 reports results across three benchmarks at k=1. Unlike direct single-pass answering, our

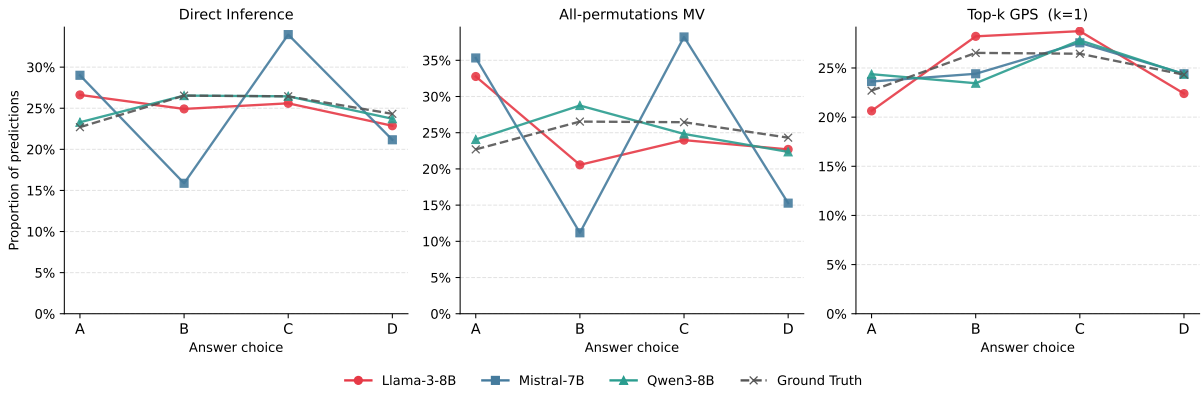

Figure 4: Predicted label distributions on **ARC Challenge** (4 choices) under three conditions: direct inference, all-permutations majority voting (MV), and Top-k GPS (k=1). The dashed gray line shows the near-uniform gold label distribution. GPS moves all three models closer to the gold reference.

method routes high-sensitivity questions to a scorer-selected permutation while retaining the original-order prediction for low-sensitivity questions. We compare against two baselines: the standard single-pass **Baseline** (original order), and a **Random** permutation selected from a uniform distribution of the $N!$ orderings. Random results are averaged over three runs.

Accuracy and Macro F1. Our method improves consistently across all nine model-dataset combinations, with average accuracy gains of $+1.32\%$ on MMLU, $+2.73\%$ on CommonsenseQA, and $+1.62\%$ on ARC Challenge. The gains are most pronounced for Mistral-7B, which improves by $+2.52\%$, $+6.22\%$, and $+2.38\%$ respectively, a direct consequence of its exceptionally high baseline sensitivity, as evidenced by the RSD values of 20.32%, 38.90%, and 18.74%. Similarly, Llama-3-8B improves from 76.96% to 78.24% on ARC Challenge, even with a relatively low baseline sensitivity. The random baseline, despite occasionally matching, or slightly exceeding the original-order baseline on accuracy (e.g. 77.33% vs. 76.96% for Llama-3-8B on ARC Challenge), does so inconsistently across models and datasets, confirming that any individual gain is attributable to chance rather than a principled selection mechanism. Our method, by contrast, outperforms both baseline and random on accuracy and Macro F1, across all nine combinations, demonstrating that the permutation scorer reliably identifies better-performing orderings rather than benefiting from random luck.

Bias-specific metrics. The most striking improvements appear in RSD and CKLD. RSD falls on average by 6.41, 9.28, and 3.56 percents across the three datasets, with Mistral-7B's CommonsenseQA RSD dropping from 38.90% to 22.21% (-43% relative). CKLD reduces even more dramatically, with Mistral-7B's CommonsenseQA CKLD collapsing from 19.48% to 3.95%. The random baseline occasionally reduces RSD, or CKLD on individual cells, but it also inflates it elsewhere (e.g. Mistral-7B's RSD on CommonsenseQA rises from 38.90% to 40.51%). This confirms that positional bias suppression requires a learned scoring signal. The *PermScorer* of GPS provides this by learning, from the model's own hidden-state representations, which orderings are most likely to draw a correct and position-invariant response. To ensure a stable debiasing effect, the scorer systematically surfaces permutations aligned with the model's factual representations rather than relying on chance. Three design choices underpin this: (1) the sensitivity classifier bypasses re-ranking for robust questions, focusing compute on vulnerable ones; (2) the *PermScorer* selects a reliable permutation at k=1, and (3) training on the model's own hidden-state embeddings ties the scorer to the same representations driving positional preference.

Prediction Distribution Calibration. Figures 4, 5, and 6 visualize per-choice prediction distributions across all three datasets. Models whose baseline predictions are heavily skewed toward one or two answer slots are pulled toward a more uniform distribution under GPS, aligning more closely with the gold distribution. For instance, Llama-3-8B on MMLU-Redux-2.0 shows a moderate A-preference at direct inference which MV amplifies into a sharper bias (A rises to 36.4%, B collapses to 16.7%). whereas GPS produces the flattest distri-

Metric		MMLU-Redux 2.0			CommonsenseQA			ARC Challenge		
		Llama-3-8b	Mistral-7b	Qwen3-8b	Llama-3-8b	Mistral-7b	Qwen3-8b	Llama-3-8b	Mistral-7b	Qwen3-8b
Acc% [↑]	ICL	62.29%	52.63%	**76.30%**	69.62%	57.66%	78.46%	78.92%	64.93%	90.36%
	ICL + GPS	**62.80%**	**54.22%**	75.89%	**71.42%**	**61.75%**	**79.36%**	**79.69%**	**68.34%**	**91.04%**
MacF1% [↑]	ICL	62.23%	52.00%	**76.21%**	69.44%	57.28%	78.36%	78.89%	64.90%	90.41%
	ICL + GPS	**62.73%**	**53.89%**	75.79%	**71.33%**	**61.52%**	**79.24%**	**79.68%**	**68.28%**	**91.06%**
RSD% [↓]	ICL	5.58%	24.30%	**2.26%**	8.29%	21.67%	4.59%	3.24%	17.76%	1.41%
	ICL + GPS	**2.77%**	**16.02%**	2.67%	**5.10%**	**15.63%**	**4.04%**	**1.39%**	**12.66%**	**1.07%**
CKLD% [↓]	ICL	0.97%	6.34%	**0.04%**	0.51%	7.00%	0.28%	0.21%	4.31%	0.10%
	ICL + GPS	**0.41%**	**2.31%**	0.06%	**0.19%**	**3.47%**	**0.13%**	**0.04%**	**2.05%**	**0.02%**

Table 2: ICL vs. ICL + GPS results across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge datasets. GPS improves over ICL on most combinations, with the largest gain on Mistral-7B's CKLD on MMLU.

Metric		MMLU-Redux 2.0			CommonsenseQA			ARC Challenge		
		Llama-3-8b	Mistral-7b	Qwen3-8b	Llama-3-8b	Mistral-7b	Qwen3-8b	Llama-3-8b	Mistral-7b	Qwen3-8b
Acc% [↑]	CoT	62.66%	52.89%	75.61%	71.90%	56.59%	**77.97%**	77.90%	67.83%	90.61%
	CoT + GPS	**62.96%**	**55.89%**	**75.98%**	**72.97%**	**59.71%**	77.72%	**79.18%**	**70.99%**	**92.41%**
MacF1% [↑]	CoT	62.65%	52.58%	75.54%	71.97%	55.52%	**77.79%**	77.89%	67.73%	90.67%
	CoT + GPS	**62.94%**	**55.76%**	**75.91%**	**72.93%**	**59.17%**	77.61%	**79.17%**	**70.98%**	**92.41%**
RSD% [↓]	CoT	8.81%	22.77%	**1.16%**	**6.35%**	31.60%	10.88%	5.63%	14.72%	2.73%
	CoT + GPS	**5.44%**	**8.12%**	1.25%	6.44%	**22.03%**	**9.08%**	**2.52%**	**6.62%**	**1.33%**
CKLD% [↓]	CoT	1.78%	6.74%	0.11%	0.41%	11.55%	1.16%	0.64%	2.76%	0.09%
	CoT + GPS	**0.84%**	**0.68%**	**0.05%**	**0.38%**	**4.43%**	**0.83%**	**0.14%**	**0.37%**	**0.02%**

Table 3: CoT vs. CoT + GPS results across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge datasets. CoT + GPS improves on majority combinations, notably reducing CKLD in every case.

bution with all choices between 23-26%. GPS actively de-concentrates positional preference rather than merely shifting errors across slots.

6 Discussion

We examine four questions in this section. **(1)** Is GPS complementary to existing prompting-based strategies such as ICL, and CoT? **(2)** Does the reverse-order meaningfully contribute over the identity-order? **(3)** Does GPS generalize across domains and model families without retraining? **(4)** Does the debiasing effect extend to other inference-based QA tasks?

6.1 Complementary Baselines

In-Context Learning. We evaluate our method on top of a 3-shot In-Context Learning prompt (Table 2), where three labeled examples are prepended to each question before inference (see Appendix L). ICL alone partially reduces positional sensitivity by providing answer format anchors, yet positional bias persists. Mistral-7B retains an RSD of 17.76% on ARC Challenge, and 21.67% on CommonsenseQA under ICL. Applying GPS on top (k=1) improves accuracy on most model-dataset combinations, and consistently reduces both RSD and CKLD. The largest gains are observed for Mistral-7B (e.g. CKLD from 7.00% to 3.47% on CommonsenseQA). The single drop for Qwen3-8B on MMLU stems from its near-zero permutation sensitivity on MMLU (median $= 0.055$; $\sim$49% of samples have sensitivity < 0.05), leaving the scorer with insufficient signal to outperform the identity ordering on the routed high-sensitivity samples.

Chain-of-Thought (CoT). For the CoT baseline, GPS generates an intermediate reasoning chain of 50-100 tokens before committing to an answer letter. Despite generating explicit reasoning, CoT does not eliminate positional bias. Mistral-7B shows RSD of 22.77% on MMLU and 31.60% on CommonsenseQA under CoT alone. CoT + GPS (k=1) improves over CoT on accuracy for 8 of 9 combinations and reduces CKLD across the board, most notably collapsing Mistral-7B's CKLD from 6.74% to 0.68% on MMLU-Redux-2.0. Gains on MMLU-Redux 2.0 are comparatively modest, which we attribute in part to the larger test set (5,330 samples) being evaluated with only 50 reasoning tokens; we treat a thorough token-budget study as future work. Together, these results show

that GPS is complementary to prompting-based bias mitigation strategies (see Appendix L).

		Llama-3-8b	Mistral-7b
ARC Challenge (MMLU→Arc)	Baseline	76.96%	64.50%
	GPS	**78.24%**	66.89%
	Cross Domain	77.21%	**68.77%**
MMLU-Redux (Arc→MMLU)	Baseline	63.38%	54.54%
	GPS	64.39%	**57.05%**
	Cross Domain	**64.99%**	56.04%

Table 4: Cross-domain inference results (K=1, 4 options) without target-domain fine-tuning.

6.2 Identity vs. Identity+Reverse Order

The identity-permutation variant estimates sensitivity using a single forward pass over the original option ordering, omitting the two-pass that GPS uses to average out positional bias. Table 12 compares these two variants of GPS. For near-unbiased models such as Qwen3-8B, a single forward pass occasionally suffices, as the hidden state alone captures most of the available signal. However, for high-bias models such as Mistral-7B, the direct pass variant consistently underperforms (e.g. 55.86% vs. 59.95% on CommonsenseQA). Results show that, the reverse pass is not an auxiliary step but a definitional one. The full two-pass *PermScorer* outperforms the identity variant on most cases.

6.3 Cross Domain Analysis

Table 4 reports a cross-domain setting where both the sensitivity classifier and the permutation scorer are trained on one dataset and applied to another without any target-domain fine-tuning. MMLU-Redux-2.0 and ARC Challenge datasets are directly compatible for this experiment, since both use four answer options. In all four model-dataset combinations, the cross-domain variant still outperforms the baseline, demonstrating that permutation sensitivity is a model-level property rather than a dataset-specific artifact. The gains are competitive with the in-domain method: cross-domain even surpasses it for Mistral-7B on ARC Challenge (68.77% vs. 66.89%) and for Llama-3-8B on MMLU (64.99% vs. 64.39%).

6.4 Experiments : Additional Models, Dataset

GPS generalizes cleanly to sub-8B models G. Llama-3.2-3B (Grattafiori et al., 2024) improves from 66.81% to 69.62% on ARC Challenge (RSD: 6.91% → 1.74%), and Gemma3-4B (DeepMind, 2025) shows consistent gains across all three benchmarks, including a substantial bias reduction.

Within a model family. Comparing Gemma-7b, Gemma3-4B, and Gemma3-12B (Team and Google, 2024; DeepMind, 2025) reveals a consistent pattern across model generations, and capacities. Our method improves accuracy and considerably reduces bias across all three variants. On CommonsenseQA, accuracy improves by +2.62%, +2.13%, and +1.97% for Gemma-7b, Gemma3-4B, and Gemma3-12B respectively, while CKLD reduces from 5.09% to 0.95%, 2.57% to 0.31%, and 1.46% to 0.31%, with a consistent reduction in RSD as well. The larger model shows lower baseline bias, so absolute accuracy gains are smaller, but bias suppression remains stable across the family, indicating the pipeline adapts to each model's operating point without re-calibration.

Inference-based QA Task. We evaluate on HellaSwag (Zellers et al., 2019), a sentence-completion task requiring the model to select the most plausible continuation from four candidates (Table 11). GPS improves accuracy for both Llama-3-8B (66.28% → 67.29%), and Mistral-7B (47.06% → 48.07%), with strong debiasing for all three models. It indicates that positional preference is a general artifact of auto-regressive scoring that our method suppresses regardless of task type.

7 Conclusion

In this paper, we present **GPS** (Guided Permutation Scoring), a novel approach that permutes option-text order in MCQA without modifying the base LLM. By combining a small sensitivity classifier with a learned *PermScorer* trained on the model's own hidden-state representations, GPS selectively applies permutation-based correction only where it is needed, and identifies top-k option orderings among all candidates. At k=1, GPS consistently improves both accuracy and Macro F1 across all model-dataset combinations, while producing a stable reduction in positional bias as measured by RSD and CKLD. Additionally, GPS transfers across datasets without target-domain fine-tuning, and is complementary to prompting strategies such as ICL and CoT. Collectively, these properties position GPS as a practical, broadly applicable debiasing component that can be used on top of any frozen LLM out of the box.

Limitations

While GPS demonstrates consistent gains across diverse models and benchmarks, three limitations remain. First, GPS is not directly assessed on large-scale thinking models (e.g., o1, o3 (OpenAI, 2024, 2025)), primarily due to practical constraints. Each permutation pass triggers a full extended reasoning trace making API calls costly, and such models typically restrict access to hidden-state representations, rendering the *PermScorer*'s reliance on internal embeddings inapplicable in these settings. Second, GPS inherently requires two LLM forward passes per sample to extract preliminary confidence features, and while this overhead is constant and independent of k, finding a more efficient alternative to this feature extraction phase remains an open challenge for future work. Third, while increasing k monotonically closes the gap with exhaustive majority voting, the minimum k at which GPS is guaranteed to match or surpass it lacks a principled characterization, and identifying this crossover point is a promising future direction.

References

Hyeong Kyu Choi, Weijie Xu, Chi Xue, Stephanie Eckman, and Chandan K. Reddy. 2025. Mitigating selection bias with node pruning and auxiliary options. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5190–5215, Vienna, Austria. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Google DeepMind. 2025. Gemma 3 technical report.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. 2025. Are we done with mmlu? *Preprint*, arXiv:2406.04127.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Blessed Guda, Lawrence Francis, Gabrial Zencha Ashungafac, Carlee Joe-Wong, and Moise Busogi. 2025. Quantifying and mitigating selection bias in LLMs: A transferable LoRA fine-tuning and efficient majority voting approach. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2019–2038, Mumbai, India. The Asian Federation of Natural Language Processing and The 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, Lava Kumar Renamik, Pierre-Emmanuel Richemond, Pedro Rodriguez, Thibaut Sellam, Teven Le Scao, William Teven, William Marshall, Timothée Lacroix, and Guillaume Artetxe. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. 2025. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16537–16552.

Adian Liusie, Yassir Fathullah, and Mark Gales. 2024. Teacher-student training for debiasing: General permutation debiasing for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1376–1387.

OpenAI. 2024. OpenAI o1 system card. `https://arxiv.org/abs/2412.16720`. *Preprint*, arXiv:2412.16720.

OpenAI. 2025. Introducing o3 and o4-mini. `https://openai.com`. Accessed: 2026-05-23.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017.

Yuval Reif and Roy Schwartz. 2024. Beyond performance: Quantifying and mitigating label bias in llms. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6784–6798.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. *Preprint*, arXiv:2210.12353.

A. Saltelli. 2008. *Global sensitivity analysis: the primer*. John Wiley.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemini Team and Google. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Sheng-Lun Wei, Cheng-Kuang Wu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. Unveiling selection biases: Exploring order and token sensitivity in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5598–5621.

Mengge Xue, Zhenyu Hu, Liqun Liu, Kuo Liao, Shu'ang Li, Honglin Han, Meng Zhao, and Chengguo Yin. 2024. Strengthened symbol binding makes large language models reliable multiple-choice selectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4344.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations*, volume 2024, pages 19426–19454.

Jinquan Zheng, Jia Yuan, Jiacheng Yao, Chenyang Gu, Pujun Zheng, and Guoxiu He. 2026. Mitigating selection bias in large language models via permutation-aware grpo. *arXiv preprint arXiv:2603.21016*.

Hanzhang Zhou, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. 2024. Unibias: Unveiling and mitigating llm bias through internal attention and ffn manipulation. *Advances in Neural Information Processing Systems*, 37:102173–102196.

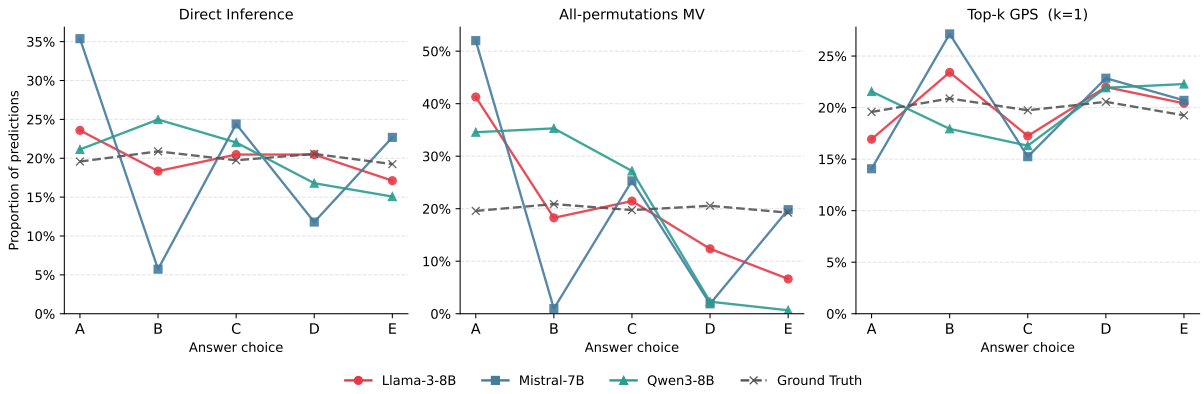

Figure 5: Predicted label distributions on **CommonsenseQA** (5 choices). All-permutations MV produces the most extreme skews across datasets. Qwen3-8B nearly abandons choices D and E, and Llama-3-8B concentrates over 41% of its predictions on choice A. GPS (k=1) recovers a near-uniform spread across all five options for both models, closely tracking the gold label distribution.

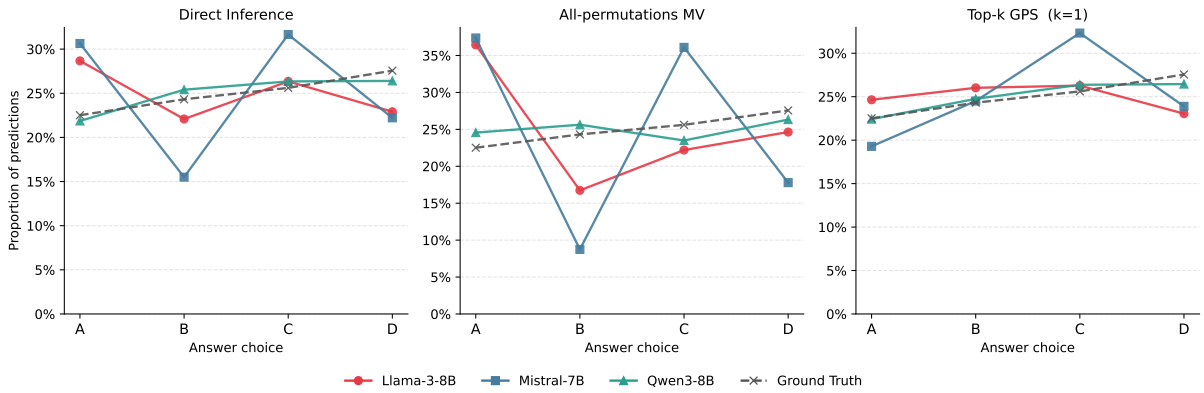

Figure 6: Predicted label distributions on **MMLU-Redux-2.0** (4 choices). All-permutations MV amplifies existing positional biases: Llama-3-8B's moderate A-preference sharpens, while Mistral-7B develops a severe A/C spike. Top-k GPS (k=1) corrects both, yielding the flattest per-model distributions across all three conditions and aligning closely with the gold reference.

A Scalar Features used in the Sensitivity Classifier

The sensitivity classifier is conditioned on ten scalar features derived directly from the model's output distributions. These features capture complementary uncertainty signals that are not easily recoverable from the compressed embedding projection alone.

Forward-pass confidence (e_1, m_1, $p_1^{\max}$): the Shannon entropy, top-two probability margin, and maximum softmax probability computed over the option logits of the canonical (identity) ordering. High entropy or low margin indicates that the model is already uncertain in its default ordering, which is a strong prior signal for high permutation sensitivity.

Reverse-pass confidence. (e_2, m_2, $p_2^{\max}$): the same three statistics computed under the fully reversed option ordering. These directly quantify how much the model's confidence profile changes when the option order is flipped.

Cross-pass agreement and deltas (δ_e, δ_m, agree, $p^{\min}$): the difference in entropy ($\delta_e = e_1 - e_2$) and margin ($\delta_m = m_1 - m_2$) between the two passes; a binary flag indicating whether both passes predict the same answer letter; and the minimum max-probability across the two passes. A large $|\delta_e|$ or $|\delta_m|$, or a disagreement between passes, is a direct indicator that the model's decision is unstable under reordering, precisely the condition that warrants routing to the top-k MV path.

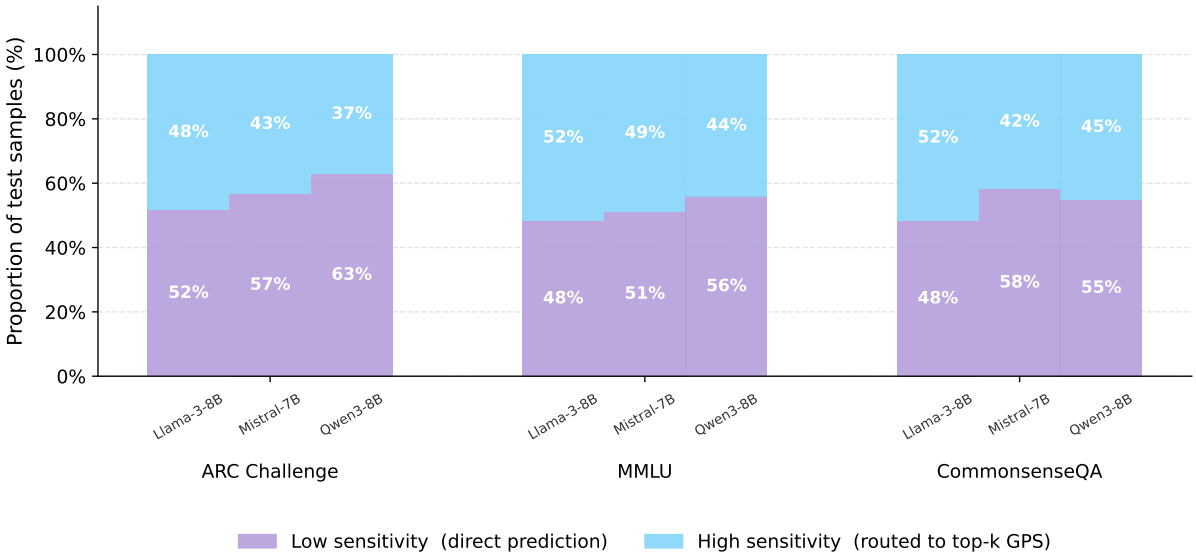

Figure 7: Sensitivity classifier routing ratios across datasets and models. Each stacked bar shows the proportion of test samples classified as low-sensitivity (green; answered via direct prediction) versus high-sensitivity (red; routed to top-k GPS, k=1). The classifier partitions samples roughly evenly across all settings, with the high-sensitivity share ranging from 37% to 52%, confirming that neither path dominates and the classifier makes non-trivial routing decisions for every model–dataset combination. Qwen3-8B consistently receives the lowest high-sensitivity ratio (37-45%), suggesting its predictions are more permutation-stable, while Llama-3-8B is routed to GPS most aggressively on MMLU, and CommonsenseQA (52% each).

B Sensitivity Classifier Routing Ratios Across Models and Datasets

Figure 7 breaks down the proportion of test samples routed to each path by the sensitivity classifier across all three models, and three datasets. The high-sensitivity share ranges from 37% to 52%, confirming that the classifier makes non-trivial routing decisions in every model-dataset combination.

C Permutation Sensitivity Characterization

Before building the training data to train the Sensitivity Classifier, we analyze the sensitivity of an LLM on each dataset. For the datasets with $N = 4$ option-choices, we use all the $N! = 24$ permutations of the option choices in the LLM prompt during the data preprocessing step. But, for CommonsenseQA with $N = 5, N! = 120$, we run for a configuration of 50 permutations of the choices. For each training sample $(q, \mathcal{O}, y^*)$, we run the model on the respective permutations, and record

$$p_{\text{correct}}(\pi_i) = \text{softmax}(\mathbf{z}(\pi_i))_{\pi_i^{-1}(y^*)}, \quad (7)$$

for $i = 1, \ldots, N!$, where $\mathbf{z}(\pi_i) \in R^N$ contains the option-letter logits under permutation π_i, and $\pi_i^{-1}(y^*)$ is the position of the gold answer after reordering. Thus p_{correct} is the softmax probability assigned to the gold answer under π_i. The *permutation sensitivity* of a sample is defined as

$$\sigma_q = \text{std}\left(\{p_{\text{correct}}(\pi_i)\}_{i=1}^{N!}\right). \quad (8)$$

This formulation is motivated by variance-based sensitivity analysis (Saltelli, 2008): if a model relies purely on answer semantics, p_{correct} should remain stable across permutations, yielding $\sigma_q \approx 0$; a large σ_q indicates that the model's confidence in the correct answer fluctuates with slot position, which is precisely the signature of positional bias. The standard deviation across permutations thus serves as a model-agnostic, label-aware probe of how much a given question exposes the model's positional preference, consistent with prior observations that LLMs exhibit significant sensitivity to answer ordering (Pezeshkpour and Hruschka, 2024).

Threshold selection. We use the *median* of $\{\sigma_q\}$ over the training set as the decision boundary for the binary sensitivity classifier. This choice is deliberate: the median is inherently robust to the long-tailed distribution of sensitivity values that typically arises (a small fraction of samples exhibit extreme sensitivity while most cluster near zero), and it

yields a balanced 50/50 label split that prevents class-imbalance artifacts during classifier training.

Four-regime analysis and its connection to the median threshold. $n_{\text{correct}} = \sum_i \mathbf{1}[f_\theta(\pi_i; q) = y^*]$ is basically the count of how many times among all the permutation orderings the predicted letter choice has been the actual ground truth label. To understand what the sensitivity score σ_q captures, We partition the data samples into four mutually exclusive regimes based on their permutation behavior (Table 5), we assign each sample to one of four regimes. Here $t_1 = 12, 25$ for $N = 4$, and $t_2 = 20, 42$ for $N = 5$.

- **Regime 1** ($n_{\text{correct}} \geq t_2$): correct on nearly all permutations, robust to order (low σ_q, direct prediction reliable).
- **Regime 2** ($t_1 \leq n_{\text{correct}} < t_2$): majority vote is correct but individual passes vary.
- **Regime 3** ($n_{\text{correct}} < t_1$, MV correct): a minority of passes are correct, yet the majority vote recovers the right answer.
- **Regime 4** ($n_{\text{correct}} < t_1$, MV wrong): the model is fundamentally misled on this question.

Dataset	Model	R1 (%)	R2 (%)	R3 (%)	R4 (%)
ARC Challenge	Llama-3-8B	64.7	14.0	1.8	19.5
	Mistral-7B	37.8	31.1	5.5	25.5
	Qwen3-8B	84.8	6.6	0.3	8.3
MMLU-Redux 2.0	Llama-3-8B	45.1	17.0	4.7	33.2
	Mistral-7B	26.0	26.0	9.1	38.9
	Qwen3-8B	62.6	13.5	1.1	22.9
CommonsenseQA	Llama-3-8B	53.2	17.6	2.9	26.3
	Mistral-7B	16.4	41.4	13.8	28.4
	Qwen3-8B	61.6	18.9	1.6	17.9

Table 5: Percentage of test samples in each regime per model and dataset. R1: all/most permutations correct; R2: majority correct (MV helps); R3: minority correct, MV still wins; R4: MV fails.

Two patterns emerge directly from Table 5. First, Regime-3 is consistently small ($< 14\%$ everywhere), confirming that the boundary between "MV recoverable" and "MV fails" is sharp rather than gradual. So, there is a real bipartition in the data. Second, Regime 1 varies notably with the model's baseline bias : Qwen3-8B, which is already near-unbiased, has 62–85% of samples in Regime 1, while Mistral-7B has as few as 16% on CommonsenseQA. Selecting the *median* of σ_q as the classifier threshold exploits this structure directly, it positions the decision boundary inside the bulk of the distribution rather than at an extreme, routing the densest cluster of consistently-correct Regime 1 samples to direct prediction while sending the more variable Regime-2/3/4 samples to the top-k path, and it does so in a model-specific way, automatically adapting to models like Qwen that concentrate most samples in Regime 1 versus models like Mistral where the distribution is far more spread.

After dividing the data samples into either robust or highly sensitive class, we train the sensitivity classifier.

Evaluation of Sensitivity Classifier. Table 6 reports 5-fold cross-validated ROC-AUC and balanced accuracy (BAcc) across all model-dataset combinations. The classifier performs strongly on most settings, with cross-validated ROC-AUC ranging from 0.90 to 0.96 for Llama-3-8B and Qwen3-8B, and balanced accuracy consistently above 0.83. Performance is somewhat lower for Mistral-7B, likely due to its higher positional bias producing noisier confidence signals that are harder to separate. Train and cross-validation metrics are closely aligned across all settings, indicating no significant overfitting.

D Overall per-sample cost.

Let r be the fraction of samples routed to Phase 2F (i.e. predicted high-sensitivity). The expected number of LLM passes per sample is

$$E[\text{passes}] = (1-r)\cdot 2 + r\cdot(2+k) = 2 + r\,k. \quad (9)$$

Compared with 24 for exhaustive majority voting on all samples, a reduction factor of up to $24/(2+rk)$ depending on routing rate and chosen k.

E Dataset and Model Statistics

Table 7 summarizes the train/test splits and answer-choice counts used across our three benchmarks. All datasets, and models are used under their publicly available academic licenses.

Each benchmark required minor preprocessing to ensure compatibility with our evaluation framework. For **CommonsenseQA**, the official test set does not include ground-truth labels, so we use the validation split as our test set. For **ARC Challenge**, a small number of samples contained either more than four answer options, or non-standard

Dataset	Model	CV ROC-AUC	CV BAcc	Train ROC-AUC	Train BAcc
ARC Challenge	Llama-3-8B	0.901	0.830	0.905	0.830
	Mistral-7B	0.773	0.714	0.780	0.722
	Qwen3-8B	0.965	0.870	0.965	0.872
MMLU	Llama-3-8B	0.794	0.716	0.801	0.724
	Mistral-7B	0.755	0.715	0.761	0.720
	Qwen3-8B	0.897	0.833	0.900	0.833
CommonsenseQA	Llama-3-8B	0.790	0.712	0.791	0.713
	Mistral-7B	0.620	0.595	0.623	0.595
	Qwen3-8B	0.902	0.830	0.902	0.830

Table 6: Sensitivity classifier evaluation (5-fold CV) across datasets and models.

Dataset	Train	Test	# Options
ARC Challenge	1,119	1,172	4
MMLU-Redux 2.0	1,531	5,330	4
CommonsenseQA	9,741	1,221	5

Table 7: Train/test split sizes and number of answer choices per dataset.

option labels (e.g., numerals $1 \ldots 4$ instead of letters A. . .D); we remapped all numeric labels to their alphabetic equivalents, and reduced samples which had more than four option-choices to a standard four-choice format. For **MMLU**, we use the original MMLU split for training and evaluate on MMLU-Redux 2.0, aggregating examples across all subjects and retaining only those annotated with `error_type = "ok"`, which identifies questions that were verified to be correctly formulated and unambiguous.

The Table 8 lists all the models used in our experiments, along with their parameter count.

Model	Parameters
LLaMA-3-8B-Instruct	8.0B
LLaMA-3.2-3B-Instruct	3.2B
Gemma–7B	8.5B
Gemma-3-4B-IT	4.0B
Gemma-3-12B-IT	12.0B
Qwen3-8B	8.0B
Mistral-7B-Instruct-v0.2	7.2B

Table 8: LLMs used in our experiments, with their total parameter counts.

F Hyperparameters

Sensitivity Classifier. The sensitivity classifier is a logistic regression model trained on a 10-dimensional feature vector extracted from two LLM passes (identity, and reverse option order). Labels are binarized at the median permutation-sensitivity of the training set (1 = HIGH, 0 = LOW). The model uses ℓ_2 regularization ($C = 1.0$), balanced class weights, and is fit via L-BFGS with a maximum of 1 000 iterations. Input features are standardized with `StandardScaler` before training. Generalization is estimated via 5-fold stratified cross-validation.

Permutation Scorer. The permutation scorer is a lightweight neural network that assigns a scalar score to each of the possible answer-option orderings. The architecture consists of three components: (i) a shared linear projection opt_proj : $R^d \to R^{64}$ applied uniformly to input token embeddings, and to the hidden states from both LLM passes (all three having dimension $d = d_{\text{model}}$); (ii) a context encoder ctx_proj : $R^{138} \to R^{256}$ (Linear $\to$ LayerNorm $\to$ ReLU) that fuses mean-pooled projections of the two hidden-state passes with the scalar features; and (iii) a two-layer scorer MLP $R^{512} \to R^{256} \to R^1$ (ReLU nonlinearity) that concatenates the permuted option projections with the context vector. With the hidden states of LLaMA-3-8B ($d = 4096$) this yields **429,889** trainable parameters in total. The model is trained for 50 epochs with the Adam optimizer (lr $= 10^{-3}$, gradient clipping at $\|\cdot\|_2 = 1.0$), and binary cross-entropy loss with a positive-class weight of 3.0 to compensate for the class imbalance inherent in the per-permutation correctness labels.

G Results using additional LLMs

Table 9 extends the main results to four additional models: Gemma-7B, Llama-3.2-3B, Gemma-3-4B-IT, and Gemma-3-12B-IT across all three bench-

LLMs		MMLU-Redux 2.0 (K=1)				CommonsenseQA (K=1)				ARC Challenge (K=1)			
		Acc. [↑]	**Macro F1** [↑]	**RSD** [↓]	**CKLD** [↓]	**Acc.** [↑]	**Macro F1** [↑]	**RSD** [↓]	**CKLD** [↓]	**Acc.** [↑]	**Macro F1** [↑]	**RSD** [↓]	**CKLD** [↓]
Gemma-7b	Baseline	**60.38%**	**60.38%**	2.48%	0.22%	62.41%	61.98%	19.10%	5.09%	76.19%	76.17%	3.65%	0.15%
	Random	58.71%	58.67%	**2.47%**	**0.09%**	62.46%	62.42%	19.01%	5.14%	75.57%	75.52%	2.89%	0.23%
	GPS	59.14%	59.11%	3.37%	0.41%	**65.03%**	**64.92%**	**9.06%**	**0.95%**	**76.45%**	**76.48%**	**1.69%**	**0.03%**
Llama-3.2-3b	Baseline	56.34%	56.20%	9.04%	1.43%	56.02%	54.73%	31.36%	9.24%	66.81%	66.69%	6.91%	0.38%
	Random	55.06%	54.99%	8.22%	0.90%	55.66%	55.65%	27.55%	8.24%	66.38%	66.34%	7.27%	0.18%
	GPS	**56.51%**	**56.46%**	**4.87%**	**0.89%**	**57.58%**	**57.58%**	**11.10%**	**1.68%**	**69.62%**	**69.57%**	**1.74%**	**0.15%**
Gemma3-4b-it	Baseline	54.39%	54.35%	16.80%	5.94%	65.68%	65.42%	15.84%	2.57%	72.01%	72.13%	8.78%	1.41%
	Random	54.22%	54.19%	17.60%	3.90%	64.70%	64.71%	10.02%	1.69%	70.73%	70.65%	11.45%	2.37%
	GPS	**56.49%**	**56.46%**	**4.40%**	**0.68%**	**67.81%**	**67.80%**	**5.87%**	**0.31%**	**72.35%**	**72.32%**	**3.98%**	**0.27%**
Gemma3-12b-it	Baseline	**73.55%**	**73.55%**	4.26%	0.65%	78.21%	77.98%	11.91%	1.46%	89.68%	89.70%	**1.66%**	0.06%
	Random	71.97%	71.94%	3.34%	**0.22%**	76.49%	76.48%	9.80%	1.66%	89.39%	89.40%	1.97%	0.06%
	GPS	73.47%	73.44%	**2.26%**	0.24%	**80.18%**	**80.10%**	**6.41%**	**0.31%**	**90.18%**	**90.18%**	1.90%	**0.03%**

Table 9: Combined results across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge (K=1) datasets for additional models.

Setting	MMLU-Redux 2.0			CommonsenseQA			ARC Challenge		
	Llama-3-8b	**Mistral-7b**	**Qwen3-8b**	**Llama-3-8b**	**Mistral-7b**	**Qwen3-8b**	**Llama-3-8b**	**Mistral-7b**	**Qwen3-8b**
Full MV	**66.83%**	**61.11%**	**77.13%**	**73.71%**	**71.58%**	**82.06%**	**80.46%**	**74.49%**	91.72%
$k = 1$	64.39%	57.04%	75.63%	70.68%	59.95%	78.21%	78.24%	66.89%	91.38%
$k = 3$	64.99%	57.52%	76.04%	71.66%	61.18%	78.38%	79.18%	70.28%	92.06%
$k = 5$	65.95%	58.71%	76.44%	72.24%	62.74%	80.02%	80.12%	71.67%	92.06%
$k = 7$	66.44%	58.87%	76.53%	72.40%	65.03%	80.67%	80.20%	71.76%	**92.15%**

Table 10: Accuracy (MicroF1) across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge datasets for various values of k and Full Majority Voting (Full MV). Full MV for CommonsenseQA was done with 50 permutations instead of 5!=120.

LLMs	Hellaswag Dataset							
	Accuracy (MicroF1) [↑]		**Macro F1** [↑]		**RSD** [↓]		**CKLD** [↓]	
	Baseline	**Ours**	**Baseline**	**Ours**	**Baseline**	**Ours**	**Baseline**	**Ours**
Llama-3-8b-instruct	66.28%	**67.29%**	66.29%	**67.30%**	6.04%	**3.59%**	0.78%	**0.29%**
Mistral-7b	47.06%	**48.07%**	46.30%	**47.94%**	37.83%	**12.52%**	15.46%	**1.46%**
Qwen3-8b	**81.12%**	81.11%	81.08%	**81.11%**	3.66%	**0.90%**	0.12%	**0.00%**

Table 11: Results on the Hellaswag Dataset.

marks. GPS consistently reduces positional bias (RSD, CKLD). It also improves, or matches accuracy, with the debiasing gains being most pronounced for the smaller, and more permutation-sensitive models.

H Impact of Permutation Budget k

Increasing k yields further consistent gains across all three benchmarks (see Table 10). On MMLU-Redux-2.0, Llama-3-8B improves from 64.39% at k=1 to 66.44% at k=7, and Mistral-7B from 57.04% to 58.87%. The trend is most pronounced on CommonsenseQA, where Mistral-7B rises from 59.95% (k=1) to 65.03% (k=7), and Qwen3-8B from 78.21% to 80.67%. This monotonic improvement confirms that the permutation scorer learns a meaningful ranking. Using top-$k = 7$ permutations narrows down the gap with majority voting in most cases. This is a testament of perm-scorer to being an efficient replacement of exhaustive all permutations approaches without.

I Results on the Hellaswag Dataset

Table 11 reports full per-model accuracy, and debiasing scores on HellaSwag Dataset under all evaluated methods. Results for Llama-3-8B, Mistral-7B, and Qwen-3-8B are included.

LLMs	MMLU-Redux 2.0		CommonsenseQA		ARC Challenge	
	Identity-GPS	**Full GPS**	**Identity-GPS**	**Full GPS**	**Identity-GPS**	**Full GPS**
Llama-3-8b-instruct	64.39%	**64.39%**	69.45%	**70.68%**	77.82%	**78.24%**
Mistral-7b	55.95%	**57.04%**	55.86%	**59.95%**	66.47%	**66.89%**
Qwen3-8b	**75.68%**	75.63%	**79.44%**	78.21%	90.53%	**91.38%**

Table 12: Ablation study results across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge datasets, comparing between Identity-Order pass, and Two-pass versions of GPS.

J Identity-Only vs. Identity+Reverse Order

For models with already low positional bias, such as Qwen3-8B, the forward-only variant occasionally edges ahead slightly: 75.68% vs. 75.63% on MMLU and 79.44% vs. 78.21% on CommonsenseQA, because the forward hidden state alone captures most of the available signal when the model is already near-unbiased. For Mistral-7B, which exhibits high baseline positional sensitivity, the forward-only variant consistently underperforms across all three benchmarks (e.g. 55.86% vs. 59.95% on CommonsenseQA), and Llama-3-8B follows the same pattern.

It is noticed that a single forward pass reveals where the model places its confidence, but only the *delta* between the identity and the reverse passes reveals whether that confidence is anchored to answer semantics or to slot position. Relying on a single pass risks misclassifying borderline samples : routing easy questions through unnecessary re-ranking or leaving hard ones in the direct-prediction path. The full two-pass estimator eliminates this inconsistency. More in Table 12.

K Token Embeddings as feature vectors

We evaluate a token-embedding-only (TE) variant as the lowest-cost alternative, replacing the two-pass hidden-state features with static input token embeddings. We report results on MMLU-Redux 2.0, whose broad subject coverage (57 subjects) makes it a representative testbed for assessing consistency across diverse question types. The results are inconsistent across models (Table 13). TE improves slightly over the baseline for Llama-3-8B (63.36% $\rightarrow$ 63.94%), yet degrades it for both Qwen3-8B (75.22% $\rightarrow$ 74.82%), and Mistral-7B (54.52% $\rightarrow$ 53.43%). This pattern is consistent across MMLU-Redux 2.0's subjects, suggesting that static token embeddings lack the contextual sensitivity signal needed to reliably distinguish high- from low-sensitivity samples.

L ICL and CoT : Prompt Templates, and Implementation Details

The ICL baseline prepends three training examples to every query. Examples are selected from the training split by finding questions for which the model predicts correctly across at least three distinct permutations of the options, then choosing a trio that covers three distinct gold-answer positions to avoid positional shortcuts.

```
prompt = f"""Output the correct answer letter
↪  option from the given choices.

Question: {ex1_question}
A: {ex1_opts[1]} B: {ex1_opts[2]} C:
↪  {ex1_opts[3]} D: {ex1_opts[4]}
Answer: {ex1_answer}

Question: {ex2_question}
A: {ex2_opts[1]} B: {ex2_opts[2]} C:
↪  {ex2_opts[3]} D: {ex2_opts[4]}
Answer: {ex2_answer}

Question: {ex3_question}
A: {ex3_opts[1]} B: {ex3_opts[2]} C:
↪  {ex3_opts[3]} D: {ex3_opts[4]}
Answer: {ex3_answer}

Question: {question}
A: {options[1]} B: {options[2]} C: {options[3]}
↪  D: {options[4]}
Answer: """
```

Answer extraction follows the same logit-based procedure as the direct baseline, combining plain and space-prefixed token ids via `logaddexp`.

The CoT baseline uses a two-pass procedure. **Pass 1** prompts the model to reason before committing to an answer:

```
# Pass 1 — generate reasoning
prompt_p1 = f"""Output the correct answer
↪  letter option from the given choices. \
Please think step by step.
```

TE Only Results	MMLU-Redux 2.0 Dataset											
	Accuracy (MicroF1) [↑]			Macro F1 [↑]			RSD [↓]			CKLD [↓]		
	Baseline	TE	Ours	Baseline	TE	Ours	Baseline	TE	Ours	Baseline	TE	Ours
Llama-3-8b-instruct	63.36%	63.94%	**64.39%**	63.33%	63.84%	**64.31%**	6.10%	**1.41%**	2.03%	1.28%	**0.11%**	0.30%
Qwen3-8b	75.22%	74.82%	**75.63%**	75.13%	74.73%	**75.55%**	1.80%	2.61%	**1.76%**	0.07%	0.03%	**0.02%**
Mistral-7b	54.52%	53.43%	**57.04%**	53.97%	53.38%	**56.97%**	20.32%	**2.02%**	5.19%	4.54%	**0.10%**	0.18%

Table 13: Token Embedding based results on MMLU-Redux 2.0 Dataset.

```
Question: {question}
A: {options[1]} B: {options[2]} C: {options[3]}
↪  D: {options[4]}
Answer: Let's think step by step."""

explanation = model.generate(prompt_p1,
↪  max_new_tokens=...)

# Pass 2 - extract answer letter from logits
prompt_p2 = prompt_p1 + explanation + "\n\nSo,
↪  the Answer is: "
pred_letter = argmax_over_option_logits(model,
↪  prompt_p2)
```

Integration with the sensitivity-routing pipeline. In the full pipeline, CoT generation occurs *after* the permutation scorer selects the top-k permutation for a HIGH-sensitivity sample. The sensitivity classifier operates on the same two-pass scalar features (entropy, margin, agreement) derived from the CoT-conditioned prompts, so explanation generation is confined to the HIGH-sensitivity branch and does not affect routing decisions for LOW-sensitivity samples.

Computational overhead. Each CoT call requires an autoregressive generation step (pass-1) in addition to the standard single forward pass used for answer extraction (pass-2). This roughly doubles the wall-clock time per sample compared to direct inference. We initially allocated a budget of 50 new tokens for pass-1; however, this was occasionally insufficient for the model to complete a coherent reasoning step, so the budget was increased to 100 tokens for the respective models to obtain reliable answer extractions. The optimal token budget is a variable worth exploring in future.

M Result analysis of Bias Node Pruning, and Ours

Bias Node Pruning (BNP). BNP (Choi et al., 2025) mitigates positional bias by pruning bias-inducing nodes from intermediate layers. Since no codebase was released, we re-implement the method, and report results on the same models and benchmarks (Table 14). Our method outperforms BNP on accuracy across all model-dataset combinations, and achieves substantially lower CKLD in almost every case (e.g. Mistral-7B CKLD: $3.91\% \rightarrow 0.18\%$ on MMLU-Redux-2.0, $20.36\% \rightarrow 3.95\%$ on CommonsenseQA). Unlike BNP, our approach requires no weight modification, making it applicable to any frozen model out of the box. See Table 14. Reproduced BNP numbers differ from the original paper, likely due to model version discrepancies.

N LLM-generated option-text orders

As an additional baseline, we prompt the LLM to generate a preferred ordering of the answer options for each question, then re-present the options in that LLM-generated order before extracting the final answer (Table 15).

Metric		MMLU-Redux 2.0 (K=1)			CommonsenseQA (K=1)			ARC Challenge (K=1)		
		Llama-3-8b	**Mistral-7b**	**Qwen3-8b**	**Llama-3-8b**	**Mistral-7b**	**Qwen3-8b**	**Llama-3-8b**	**Mistral-7b**	**Qwen3-8b**
Acc. [↑]	BNP	63.02%	54.47%	75.27%	67.24%	53.48%	78.05%	77.47%	64.33%	90.36%
	GPS	**64.39%**	**57.04%**	**75.63%**	**70.68%**	**59.95%**	**78.21%**	**78.24%**	**66.89%**	**91.38%**
Macro F1 [↑]	BNP	62.90%	53.94%	75.20%	67.16%	51.46%	77.89%	77.42%	64.17%	90.46%
	GPS	**64.31%**	**56.97%**	**75.55%**	**70.68%**	**59.36%**	**78.20%**	**78.21%**	**66.88%**	**91.41%**
RSD [↓]	BNP	11.50%	19.31%	**1.68%**	9.99%	39.74%	11.60%	1.87%	18.18%	**1.87%**
	GPS	**2.03%**	**5.19%**	1.76%	**4.24%**	**22.21%**	**1.49%**	**1.65%**	**9.78%**	2.26%
CKLD [↓]	BNP	2.82%	3.91%	0.15%	1.81%	20.36%	1.69%	0.10%	4.53%	**0.02%**
	GPS	**0.30%**	**0.18%**	**0.02%**	**0.10%**	**3.95%**	**0.01%**	**0.02%**	**0.72%**	0.05%

Table 14: Comparison of BNP (Choi et al., 2025), and GPS across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge (K=1) datasets.

Metric		MMLU-Redux 2.0 (K=1)			CommonsenseQA (K=1)			ARC Challenge (K=1)		
		Llama-3-8b	**Mistral-7b**	**Qwen3-8b**	**Llama-3-8b**	**Mistral-7b**	**Qwen3-8b**	**Llama-3-8b**	**Mistral-7b**	**Qwen3-8b**
Acc. [↑]	Baseline	63.36%	54.52%	**75.22%**	**69.04%**	53.73%	77.89%	76.96%	64.51%	90.19%
	LLM-Order	**62.04%**	**52.40%**	74.89%	67.64%	**53.64%**	**78.29%**	**77.39%**	**64.84%**	**91.21%**
Macro F1 [↑]	Baseline	**63.33%**	**53.97%**	**75.13%**	**69.04%**	**51.83%**	**77.71%**	**76.94%**	**64.30%**	**90.28%**
	LLM-Order	57.27%	50.45%	66.58%	61.61%	50.13%	66.57%	68.41%	61.78%	80.38%
RSD [↓]	Baseline	**6.10%**	**20.32%**	**1.80%**	**5.67%**	**38.90%**	**11.22%**	**3.39%**	**18.74%**	**2.24%**
	LLM-Order	23.53%	27.71%	20.63%	23.56%	44.92%	27.09%	23.82%	25.19%	16.12%
CKLD [↓]	Baseline	**1.28%**	**4.54%**	**0.07%**	**0.67%**	**19.48%**	**1.46%**	**0.42%**	**4.84%**	**0.02%**
	LLM-Order	1.01%	5.99%	0.20%	1.90%	26.54%	1.66%	0.66%	5.01%	0.23%

Table 15: Baseline vs. LLM-generated Order across MMLU-Redux 2.0, CommonsenseQA, and ARC Challenge (K=1) datasets.