

Mitigating Selection Bias in Large Language Models via Permutation-Aware GRPO

Jinquan Zheng, Jia Yuan, Jiacheng Yao, Chenyang Gu, Pujun Zheng, Guoxiu He*

School of Economics and Management, East China Normal University

{jqzheng, yuanjia, jcyao, cygu, pjzheng}@stu.ecnu.edu.cn,

gxhe@fem.ecnu.edu.cn

Abstract

Large language models (LLMs) used for multiple-choice and pairwise evaluation tasks often exhibit selection bias due to non-semantic factors like option positions and label symbols. Existing inference-time debiasing is costly and may harm reasoning, while pointwise training ignores that the same question should yield consistent answers across permutations. To address this issue, we propose Permutation-Aware Group Relative Policy Optimization (PA-GRPO), which mitigates selection bias by enforcing permutation-consistent semantic reasoning. PA-GRPO constructs a permutation group for each instance by generating multiple candidate permutations, and optimizes the model using two complementary mechanisms: (1) cross-permutation advantage, which computes advantages relative to the mean reward over all permutations of the same instance, and (2) consistency-aware reward, which encourages the model to produce consistent decisions across different permutations. Experimental results demonstrate that PA-GRPO outperforms strong baselines across seven benchmarks, substantially reducing selection bias while maintaining high overall performance. The code is available on [GitHub](#).

1 Introduction

Large language models (LLMs) have been widely applied to multiple-choice question (MCQ) reasoning (Hendrycks et al., 2020; Achiam et al., 2023) and LLM-as-a-Judge style evaluation tasks (Zheng et al., 2023b; Chiang et al., 2024). In these discrete-choice settings, option order and label symbols (e.g., A/B/C/D) are non-semantic factors that, in principle, should not influence the model’s decisions. However, as illustrated in Figure 1, even when two candidate answers are swapped without altering their content, LLMs may still favor

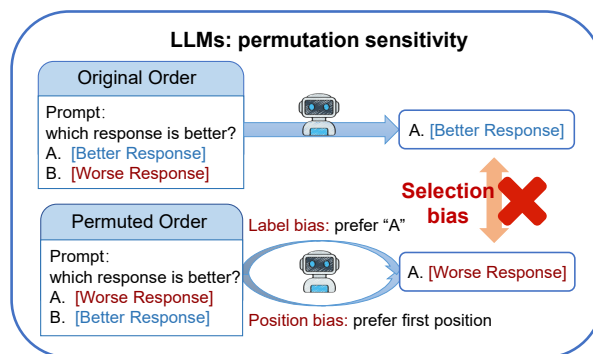


Figure 1: Example of selection bias in discrete-choice settings: swapping the order of answers changes only non-semantic factors.

the first-listed option (*i.e.*, **position bias**) or a particular label such as “A” (*i.e.*, **label bias**), leading to inconsistent decisions and even preference reversals. Such systematic deviations caused by non-semantic perturbations are referred to as **selection bias** (Zheng et al., 2023b,a). Selection bias undermines the reliability and fairness of evaluation because model outputs can depend on the arrangement of options rather than their semantic content (Shi et al., 2024; Wei et al., 2024; Wang et al., 2024a). Moreover, these errors may be further amplified in downstream applications that rely on such outputs, such as automated data construction or alignment training (Miao et al., 2024).

Existing methods for mitigating selection bias can be broadly categorized into two types: inference-time debiasing and training-time debiasing. Inference-time debiasing typically follows two routes. The first performs output-level calibration by estimating and removing priors post-hoc (Zhao et al., 2021), such as PriDe for MCQ and CalibraEval for LLM-as-a-Judge (Zheng et al., 2023a; Li et al., 2025b). While often effective, such methods merely adjust surface-level probabilities without rectifying the model’s intrinsic miscon-

*Corresponding author.

ceptions, and typically incur high computational overhead due to repeated sampling or optimization. The second route applies **internal interventions** to suppress bias sources, for instance by masking specific attention heads (UniBias) or pruning bias-related parameters (BNP) (Zhou et al., 2024; Choi et al., 2025). However, such invasive modifications complicate the inference pipeline and carry the risk of degrading general reasoning capabilities by inadvertently removing useful semantic features. In contrast, training-time debiasing aims to internalize robustness. Yet, existing supervised strategies mostly rely on static augmentation or distillation (Liusie et al., 2024; Xue et al., 2024). These methods generally treat permutations as independent static samples, limiting the model to passive imitation of the data distribution rather than actively exploring the policy space to discover robust, permutation-invariant strategies.

Fundamentally, selection bias reflects a failure of robust reasoning in discrete-choice prompting: when only non-semantic factors (labels or positions) change, the model should preserve the same semantic decision. Since this is an intrinsic reasoning flaw, a natural solution is to train the model to maintain consistency under such perturbations, rather than relying on post-hoc calibration at inference. In this context, recent reinforcement learning (RL) methods such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) are appealing, as they estimate relative advantages from multiple samples and can directly optimize decision quality.

However, standard GRPO and its variants still typically treat different prompts as independent training instances. When multiple prompt variants are derived from the same semantic instance under different permutations, they are optimized separately without an explicit **cross-permutation consistency** constraint. Consequently, the model may achieve high reward under a “favorable” permutation while performing poorly under others (e.g., reversed order), yet this inconsistency is not systematically penalized. We term this failure mode permutation-blindness.

To address this, we propose **Permutation-Aware Group Relative Policy Optimization (PA-GRPO)**, which explicitly targets **permutation invariance**: for all permuted variants of the same semantic instance, the model should make semantically consistent choices. PA-GRPO organizes multiple permutations of the same instance into a **Permutation Group** and injects permutation awareness via two

mechanisms: (1) **Cross-Permutation Advantage**: we elevate the advantage baseline from the sample mean within a single prompt to the mean over all samples across the permutation group, so that positive advantages correspond to being globally better across permutations; (2) **Consistency-Aware Reward**: we explicitly encourage consistent decisions across permutations in the reward, suppressing locally optimal strategies that exploit positional or label shortcuts.

We evaluate our approach on seven benchmarks spanning both MCQ and LLM-as-a-Judge tasks. During training, we construct representative permutation sets for each semantic instance, covering all label and position bindings, and validate on Llama-3.1-8B-Instruct, Qwen3-8B, and Qwen3-32B. Experimental results show that, compared with strong baselines such as PIF (Xue et al., 2024) and CalibraEval (Li et al., 2025b), PA-GRPO achieves superior performance in both reducing selection bias and preserving stable performance.

Our contributions are threefold:

- We identify the permutation-blindness issue in RL alignment under selection-biased settings and propose permutation invariance as a training objective.
- We propose PA-GRPO, which incorporates consistency constraints into training through a cross-permutation advantage and a consistency-aware reward.
- Extensive experiments on three backbones and seven benchmarks verify that PA-GRPO effectively reduces selection bias.

2 Related Work

LLM-as-a-Judge and MCQ. With the rapid evolution of LLMs, efficient and reliable evaluation has become a central challenge. A unifying perspective is that many widely used evaluation protocols can be formulated as discrete-choice decision making, where the model selects from a closed set of candidates. One representative line is **LLM-as-a-Judge** (Zheng et al., 2023b), which utilizes strong LLMs to perform pairwise comparisons of generated responses and has been widely adopted for alignment studies and leaderboard-style benchmarking (Li et al., 2025a). Another representative line is **Multiple Choice Question (MCQ)** evaluation (Hendrycks et al., 2020), a standard format for assessing general reasoning by constraining the output space to predefined options. Although

their application scenarios differ, both settings ultimately reduce to assigning probabilities over a closed label set (*e.g.*, A/B or A/B/C/D). Crucially, this shared discrete-choice nature makes both settings inherently susceptible to non-semantic interference: even minor perturbations to the input form can trigger drastic fluctuations in decision outcomes, undermining the reliability of such discriminative tasks (Pezeshkpour and Hruschka, 2024; Holtzman et al., 2021).

Selection Bias in LLMs. Models in discrete-choice tasks (*e.g.*, LLM-as-a-Judge, MCQ) are sensitive to non-semantic factors, known as selection bias, including label bias and position bias. **Label bias**, preferring specific IDs, is addressed by approaches like SDC (Wang and Liu, 2025), LOOC (Reif and Schwartz, 2024), and ICC (Jang et al., 2024) via content-free prompts or calibration, though often incurring extra inference overhead. **Position bias** (Lu et al., 2022), preferring specific positions, is mitigated by mechanism adjustments: PINE (Wang et al., 2024b) and RoToR (Yoon et al., 2025) modify attention masks or positional IDs, while fine-tuning strategies (Wang et al., 2025) reduce sensitivity during training.

Mitigation strategies for selection bias fall into two categories. **Inference-time debiasing** includes calibration methods like PriDe (Zheng et al., 2023a) and CalibraEval (Li et al., 2025b), which remove priors post-hoc without fixing the underlying mechanism. Alternatively, internal interventions such as UniBias (Zhou et al., 2024) and BNP (Choi et al., 2025) prune bias sources but risk unintended side effects. In contrast, **training-time debiasing** aims to internalize robustness. However, existing supervised strategies (Liusie et al., 2024; Xue et al., 2024) typically treat permutations as static samples, limiting models to passive imitation rather than actively exploring the policy space to discover robust, permutation-invariant strategies.

Reinforcement Learning for Alignment. Reinforcement Learning (RL) (Christiano et al., 2017) has become a core technology for alignment, evolving from RLHF (PPO) (Ouyang et al., 2022) to DPO (Rafailov et al., 2023), and the efficient GRPO (Shao et al., 2024) with its variants (Yu et al., 2025; Zheng et al., 2025). While these methods significantly improve generation quality, they fundamentally rely on a sample-independence assumption. Even group-based methods like GRPO typically treat permuted variants of the same semantic input as unrelated, independent events. This

leads to permutation-blindness: the optimization focuses on maximizing reward for a specific order in isolation, ignoring consistency across permutations. Consequently, existing RL frameworks fail to enforce global consistency across input variations, a critical limitation that our work seeks to address.

3 Methodology

We propose **PA-GRPO** (Permutation-Aware Group Relative Policy Optimization) to mitigate selection bias induced by non-semantic factors in discrete-choice reasoning tasks. As illustrated in **Figure 2**, unlike standard GRPO which treats each prompt variation independently, PA-GRPO explicitly models the dependency among permutation variants. Our core objective is to jointly optimize for two goals: (1) maximizing semantic correctness to align with the ground truth, and (2) enforcing permutation invariance to maintain consistent decisions regardless of option order.

3.1 Problem Formulation

A base instance x consists of a question q and a candidate set \mathcal{C} . We unify MCQ ($|\mathcal{C}| = 4$) and Pairwise Judge ($|\mathcal{C}| = 2$) as conditional generation tasks. Given a prompt p , a policy $\pi_\theta(y | p)$ generates a response y . We parse the surface label $\ell(y) \in \mathcal{L}$ (*e.g.*, A/B/C/D) from the output. Ideally, a robust model should satisfy permutation invariance: for any non-semantic permutation τ , the mapped semantic choice should remain invariant. Selection bias is defined as the violation of this property, manifested as systematic preference flips triggered by label or position changes.

3.2 Permutation Group Construction and Sampling

To enforce cross-permutation consistency, we organize training data into **Permutation Groups**. For each base instance x , we apply a set of permutation mappings Π to generate P distinct prompt variants:

$$\mathcal{G}(x) = \{p^{(t)} = \tau_t(x)\}_{t=1}^P, \quad \tau_t \in \Pi. \quad (1)$$

For each permuted prompt $p^{(t)}$, we further sample a **response group** of N outputs $\{y^{(t,i)}\}_{i=1}^N$ from the current policy π_θ (as in GRPO). Together, this yields a two-level structure: sampling diversity within each permutation and robustness across permutations. In the following sections, we compute rewards and advantage baselines over all $P \times N$

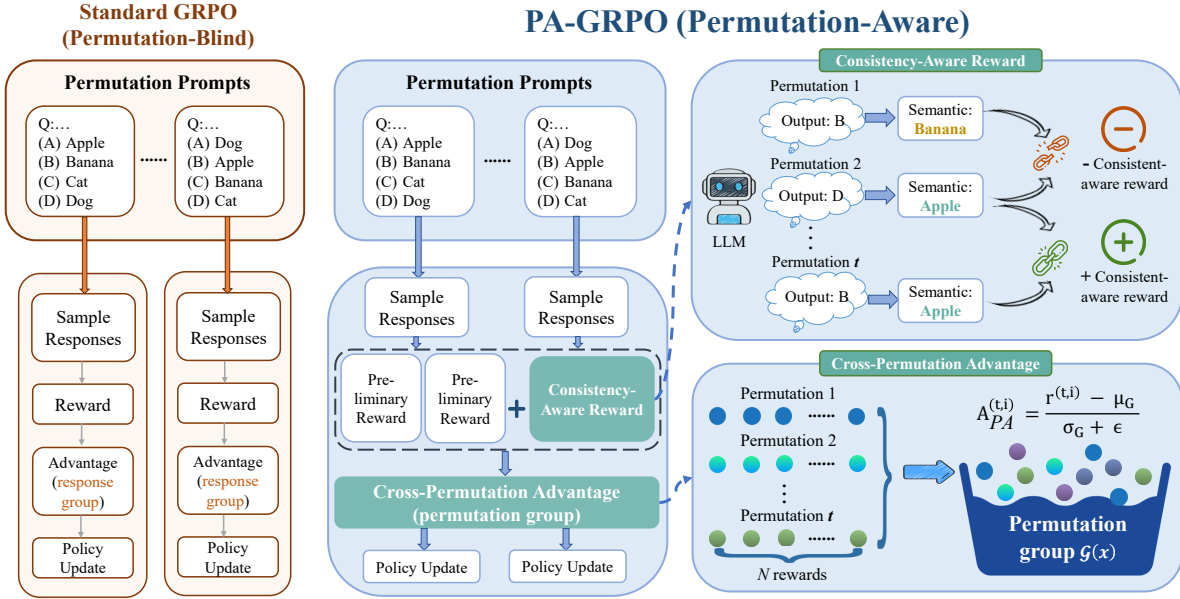


Figure 2: Comparison between **Standard GRPO** (left) and the proposed **PA-GRPO** (right). Standard GRPO treats permuted prompts as independent samples, suffering from permutation blindness where inconsistency goes unpunished. In contrast, PA-GRPO organizes samples into Permutation Groups. It introduces (1) a Cross-Permutation Advantage (using the permutation group mean as a baseline) and (2) a Consistency-Aware Reward to explicitly enforce semantic invariance across different permutations of the same instance.

samples, enabling explicit cross-permutation learning and reducing reliance on any single “favorable” permutation. Specifically, we design the permutation set Π for different tasks as follows:

MCQ ($P = 5$, **Structured Subset**). The full permutation space ($4! = 24$) is costly in computation. We employ a structured subset strategy consisting of four cyclic shifts and one reverse order:

$$\Pi_{\text{MCQ}} = \{ABCD, BCDA, CDAB, DABC, DCBA\}. \quad (2)$$

The four cyclic shifts ensure strictly uniform coverage of each option at every position. We additionally include the reverse order (DCBA) to disrupt relative adjacency patterns (e.g., A is always before B in cyclic shifts), thereby enhancing robustness against complex order-dependent biases.

JUDGE ($P = 2$, **Full Set**). For pairwise comparison, the space is small ($2! = 2$). We employ the full symmetric set:

$$\Pi_{\text{Judge}} = \{AB, BA\}. \quad (3)$$

3.3 Reward Design: Content Alignment and Consistency Awareness

Standard RLHF (Ouyang et al., 2022) typically relies on a scalar reward signal reflecting response correctness. However, in the context of debiasing, a correct response under a specific permutation is

insufficient; the model must demonstrate robust correctness across all variations. Therefore, we design a composite reward mechanism that explicitly decouples the evaluation into two dimensions: (1) **Preliminary Reward** (r_{pre}), which assesses whether the current response matches the ground truth; (2) **Consistency Reward** (r_{con}), which penalizes semantic divergence within the permutation group. To implement this, we first establish a mapping from surface labels to semantic identities.

3.3.1 Content Alignment: From Labels to Semantic Choices

Consistency must be defined over the selected content, not the surface letters. For each permutation mapping τ_t , we define a mapping function m_{τ_t} that maps a surface label to the index of the original semantic candidate:

$$m_{\tau_t}(\ell) \in \{1, \dots, |\mathcal{C}|\}. \quad (4)$$

For the i -th sampled response under permutation t , the semantic choice is:

$$z^{(t,i)} = m_{\tau_t}(\ell(y^{(t,i)})). \quad (5)$$

3.3.2 Preliminary Reward

For each sampled response $y^{(t,i)}$, the total reward is:

$$r^{(t,i)} = r_{\text{pre}}^{(t,i)} + \lambda r_{\text{con}}^{(t,i)}. \quad (6)$$

The preliminary reward aggregates accuracy, length, and format constraints:

$$r_{\text{pre}}^{(t,i)} = r_{\text{acc}}^{(t,i)} + r_{\text{len}}^{(t,i)} + r_{\text{fmt}}^{(t,i)}. \quad (7)$$

In our implementation, we use: correctness reward $r_{\text{acc}} \in \{+1, -1\}$ based on whether $z^{(t,i)}$ matches the ground-truth semantic answer, along with minor regularization terms for length ($r_{\text{len}} = \pm 0.1$) and formatting ($r_{\text{fmt}} = \pm 0.3$).

3.3.3 Consistency-Aware Reward

We introduce a consistency reward r_{con} to encourage semantically consistent decisions within each permutation group $\mathcal{G}(x)$.

JUDGE ($P = 2$): index-aligned pairwise consistency. We pair the i -th samples across the two permutations:

$$r_{\text{con}}^{(1,i)} = r_{\text{con}}^{(2,i)} = \begin{cases} +1, & z^{(1,i)} = z^{(2,i)} \\ -1, & z^{(1,i)} \neq z^{(2,i)}. \end{cases} \quad (8)$$

MCQ ($P = 5$): unique-mode agreement. Let n_k be the count of semantic choice k across the entire group, and \mathcal{M} be the set of modes:

$$n_k = \sum_{t=1}^P \sum_{i=1}^N \mathbf{1}[z^{(t,i)} = k], \quad (9)$$

$$\mathcal{M} = \arg \max_k n_k.$$

If $|\mathcal{M}| = 1$, denote the unique mode by z^* . We explicitly penalize fragmentation (including ties):

$$r_{\text{con}}^{(t,i)} = \begin{cases} +1, & |\mathcal{M}| = 1 \text{ and } z^{(t,i)} = z^* \\ -1, & \text{otherwise.} \end{cases} \quad (10)$$

3.4 Cross-Permutation Advantage Estimation

Prompts in a permutation group $\mathcal{G}(x)$ are semantically identical but differ in non-semantic surface factors. If advantages are computed independently per prompt variant (as in standard GRPO), the policy can be reinforced by a ‘‘favorable’’ ordering even when it fails under other permutations.

PA-GRPO therefore lifts advantage estimation to the **permutation-group level**. We treat the $P \times N$ samples from $\mathcal{G}(x)$ as a single comparison set and compute the permutation-group mean and standard deviation:

$$\mu_{\mathcal{G}} = \frac{1}{PN} \sum_{t=1}^P \sum_{i=1}^N r^{(t,i)}, \quad (11)$$

$$\sigma_{\mathcal{G}} = \text{Std}(\{r^{(t,i)}\}_{t,i}),$$

where both statistics are computed over all permutations and samples within $\mathcal{G}(x)$. The cross-permutation advantage is:

$$A_{\text{PA}}^{(t,i)} = \begin{cases} 0, & \sigma_{\mathcal{G}} < \delta \\ \frac{r^{(t,i)} - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \epsilon}, & \text{otherwise.} \end{cases} \quad (12)$$

This yields (i) **instance-wise competition** across permutations via the permutation-group baseline $\mu_{\mathcal{G}}$: a sample receives a positive advantage only if its reward exceeds the permutation-group mean, and (ii) **scale-invariant stability** via normalization by $\sigma_{\mathcal{G}}$. When $\sigma_{\mathcal{G}} < \delta$, rewards are nearly indistinguishable within the group, so we set $A_{\text{PA}} = 0$ to avoid amplifying noise.

3.5 Optimization Objective

We adopt a clipped objective with KL regularization to stabilize training. Let the importance ratio be defined as:

$$\rho^{(t,i)}(\theta) = \frac{\pi_{\theta}(y^{(t,i)} | p^{(t)})}{\pi_{\text{old}}(y^{(t,i)} | p^{(t)})} \quad (13)$$

The clipped surrogate objective is:

$$\mathcal{L}_{\text{clip}}(\theta) = \mathbb{E} \left[\min \left(\rho^{(t,i)} A_{\text{PA}}^{(t,i)}, \text{clip}(\rho^{(t,i)}, 1 - \eta, 1 + \eta) A_{\text{PA}}^{(t,i)} \right) \right]. \quad (14)$$

The final objective maximizes the surrogate reward while penalizing deviation from the reference policy:

$$\max_{\theta} \left(\mathcal{L}_{\text{clip}}(\theta) - \beta \mathbb{E}_{p^{(t)}} \left[\text{KL}(\pi_{\theta}(\cdot | p^{(t)}) \| \pi_{\text{ref}}(\cdot | p^{(t)})) \right] \right). \quad (15)$$

The complete training procedure is provided in **Algorithm 1** in Appendix B.

4 Experimental Setup

In this section, we present a comprehensive evaluation of PA-GRPO to answer the following research questions (RQs):

RQ1: Can PA-GRPO outperform existing inference-time calibration and training-time debiasing methods?

RQ2: What are the individual contributions of the Cross-Permutation Advantage (A_{PA}) and the Consistency-Aware Reward (r_{con})?

RQ3: How do the consistency reward coefficient λ and permutation group size P affect model performance, permutation invariance, and training efficiency?

RQ4: Can PA-GRPO effectively align reasoning consistency under Chain-of-Thought prompting?

RQ5: How does PA-GRPO’s robustness differ between isolated label-only and order-only perturbations and the naturally coupled permutations used in training, and what does this reveal about the remaining sources of bias?

4.1 Datasets and Evaluation Protocol

We evaluated PA-GRPO on seven benchmarks spanning two categories: (1) **LLM-AS-A-JUDGE**: MT-Bench (Zheng et al., 2023b), JudgeBench (Tan et al., 2024), PreferenceBench (Kim et al., 2024), and RewardBench (Lambert et al., 2025); and (2) **MCQ**: ARC-Challenge (Clark et al., 2018), TinyMMLU (Polo et al., 2024), and GPQA (Rein et al., 2024). Brief descriptions of each benchmark and their task formats are provided in Appendix C. For training, we utilized the Chatbot Arena dataset (Zheng et al., 2023b) for pairwise tasks and the MMLU training set (Hendrycks et al., 2020) for reasoning tasks. Details of data construction are provided in Appendix D.

To rigorously quantify permutation invariance during inference, we employed **Full Permutation Expansion** rather than random sampling. Specifically, we evaluated the complete permutation space for every test instance, covering all $2! = 2$ possible orders for LLM-as-a-Judge tasks and all $4! = 24$ possible arrangements for MCQ tasks.

4.2 Metrics

We report three core metrics to measure performance and robustness:

Accuracy (Acc). The standard performance metric, calculated as the average correctness over all samples in the expanded permutation groups.

Consistency (Con). It measures decision stability across permutations. For LLM-as-a-Judge, it is defined as the proportion of instance pairs where the model makes the identical semantic choice under both orders. For MCQ, we define it as the majority consensus ratio $n_{\text{mode}}/24$, where n_{mode} is the count of the most frequent semantic choice within the permutation group.

Consistent Accuracy (CA). It measures the ability to be both consistent and correct. For LLM-as-a-Judge, we apply a strict criterion: an instance

scores 1 if and only if the model answers correctly under both permutations. For MCQ, we adopt Majority Vote Correctness (Wang et al., 2022): an instance scores 1 if the mode of the permutation group matches the ground truth. This reflects whether the consensus converges to the truth after filtering out permutation noise.

4.3 Baselines

We compare PA-GRPO with five strong baselines covering both inference-time debiasing and training-time alignment. Inference-time methods include PriDe (Zheng et al., 2023a), which estimates and removes a global prior preference over option-ID tokens; CalibraEval (Li et al., 2025b), which learns a label-free calibration mapping to enforce swap-invariant judging; and UniBias (Zhou et al., 2024), which mitigates bias by masking identified biased attention heads and FFN vectors. Training-time baselines include PIF (Xue et al., 2024), an SFT algorithm that strengthens multiple-choice symbol binding via point-wise feedback on constructed negative instances, and standard GRPO (Shao et al., 2024). Due to space constraints, detailed mechanisms and our implementation settings are provided in Appendix E.

4.4 Implementation Details

We used Llama-3.1-8B-Instruct (Dubey et al., 2024), Qwen3-8B, and Qwen3-32B (Yang et al., 2025) as policy models. All models were trained within the VERL (Sheng et al., 2025) RL framework, and we employed LoRA (Hu et al., 2022) for fine-tuning. Detailed training configurations and hyperparameters are provided in Appendix F.

5 Results and Analysis

5.1 Main Results

As shown in Table 1, across three different LLMs, PA-GRPO yields consistent gains over the strongest baselines, covering both inference-time calibration and training-time debiasing methods. The improvements are particularly concentrated on permutation-sensitive robustness metrics, including Consistency and Consistent Accuracy, while keeping standard Accuracy competitive.

For **Llama-3.1-8B-Instruct**, PA-GRPO raises MT-Bench accuracy to 77.6% and consistency to 88.0%. This representative case suggests that permutation-aware optimization not only improves the headline score but also mitigates decision in-

| Model | Method | LLM-as-a-Judge | | | | | | | | | Multiple Choice Questions | | | | | | | | | | | |
|--------------|-------------|----------------|-------------|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|---------------------------|-------------|-------------|-------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| | | MT-Bench | | | JudgeBench | | | PreferenceBench | | | RewardBench | | | GPQA | | | ARC-Challenge | | | TinyMMLU | | |
| | | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA |
| Llama-3.1-8B | Base | 59.6 | 25.2 | 22.2 | 35.0 | 34.8 | 6.1 | 60.8 | 22.6 | 22.1 | 60.5 | 31.5 | 26.2 | 27.3 | 40.0 | 27.2 | 73.0 | 82.5 | 78.9 | 54.2 | 68.9 | 57.0 |
| | GRPO | 75.7 | 80.6 | 65.4 | 48.2 | 56.1 | 28.2 | 82.2 | 85.1 | 76.3 | 70.9 | 76.9 | 61.5 | 29.5 | 55.6 | 29.7 | 89.0 | 92.1 | 91.5 | 67.3 | 77.9 | 73.0 |
| | UniBias | 68.5 | 59.4 | 48.2 | 50.2 | 23.0 | 10.9 | 79.7 | 65.6 | 62.5 | 68.4 | 64.4 | 50.6 | 32.3 | 56.1 | 31.8 | 80.4 | 90.0 | 83.6 | 61.1 | 76.5 | 65.0 |
| | PriDe | 65.6 | 46.2 | 38.9 | 49.1 | 16.2 | 7.2 | 81.6 | 69.1 | 66.2 | 65.1 | 52.8 | 41.5 | 31.0 | 49.1 | 27.4 | 79.1 | 83.0 | 79.0 | 59.7 | 69.5 | 58.0 |
| | CalibraEval | 62.3 | 42.1 | 33.4 | 49.3 | 15.7 | 7.1 | 70.4 | 44.4 | 42.7 | 60.7 | 34.4 | 27.8 | 28.3 | 48.2 | 26.5 | 74.4 | 71.0 | 68.3 | 55.6 | 72.1 | 62.0 |
| | PIF | 76.1 | 84.6 | 70.4 | 53.3 | 59.2 | 30.4 | 54.3 | 37.1 | 25.7 | 73.7 | 76.7 | 62.0 | 29.1 | 49.7 | 29.2 | 73.3 | 83.0 | 78.8 | 56.9 | 69.1 | 57.0 |
| | PA-GRPO | 77.6 | 88.0 | 71.7 | 57.1 | 58.3 | 32.4 | 86.2 | 87.2 | 79.8 | 71.0 | 82.7 | 62.3 | 32.1 | 57.2 | 33.2 | 89.3 | 92.9 | 92.6 | 67.9 | 81.9 | 75.0 |
| | | (+1.5) | (+3.4) | (+1.3) | (+3.8) | (-0.9) | (+2.0) | (+4.0) | (+2.1) | (+3.5) | (-2.7) | (+5.8) | (+0.3) | (-0.2) | (+1.1) | (+1.4) | (+0.3) | (+0.8) | (+1.1) | (+0.6) | (+4.0) | (+2.0) |
| Qwen3-8B | Base | 71.2 | 56.3 | 49.4 | 43.9 | 45.5 | 16.5 | 82.3 | 70.2 | 67.4 | 79.3 | 73.8 | 66.2 | 32.5 | 48.2 | 30.4 | 90.5 | 94.7 | 92.4 | 67.4 | 79.6 | 72.0 |
| | GRPO | 78.1 | 82.5 | 69.3 | 50.4 | 62.6 | 34.8 | 82.9 | 77.3 | 76.2 | 86.8 | 86.7 | 80.2 | 40.4 | 63.2 | 43.8 | 94.6 | 97.2 | 95.9 | 56.6 | 83.7 | 76.0 |
| | UniBias | 67.0 | 52.7 | 43.4 | 52.2 | 26.1 | 14.9 | 75.8 | 55.4 | 53.5 | 66.6 | 45.9 | 39.5 | 34.6 | 54.3 | 34.2 | 89.9 | 94.1 | 91.8 | 71.3 | 81.0 | 74.0 |
| | PriDe | 72.6 | 66.7 | 52.1 | 51.2 | 48.8 | 29.8 | 81.5 | 72.6 | 69.7 | 80.9 | 79.1 | 68.7 | 34.4 | 56.9 | 38.5 | 91.2 | 94.9 | 92.6 | 68.9 | 79.9 | 78.0 |
| | CalibraEval | 72.4 | 62.1 | 50.9 | 49.7 | 56.4 | 31.3 | 76.5 | 72.8 | 70.1 | 80.4 | 71.2 | 69.5 | 33.5 | 49.8 | 32.9 | 90.1 | 91.0 | 89.8 | 63.5 | 75.5 | 68.0 |
| | PIF | 75.1 | 83.2 | 68.9 | 54.3 | 59.6 | 37.4 | 77.7 | 64.2 | 58.9 | 83.9 | 80.1 | 75.8 | 32.7 | 68.7 | 46.6 | 92.2 | 95.4 | 93.5 | 75.1 | 81.4 | 75.0 |
| | PA-GRPO | 78.8 | 86.4 | 72.0 | 60.1 | 70.0 | 45.3 | 88.9 | 89.6 | 83.7 | 84.4 | 88.9 | 80.6 | 42.3 | 72.4 | 56.7 | 95.0 | 97.5 | 96.2 | 73.2 | 88.6 | 80.0 |
| | | (+0.7) | (+3.2) | (+2.7) | (+5.8) | (+7.4) | (+7.9) | (+6.0) | (+12.3) | (+7.5) | (-2.4) | (+2.2) | (+0.4) | (+1.9) | (+3.7) | (+0.1) | (+0.4) | (+0.3) | (+0.3) | (-1.9) | (+4.9) | (+2.0) |
| Qwen3-32B | Base | 78.5 | 83.4 | 70.3 | 55.4 | 62.1 | 29.7 | 84.9 | 85.1 | 78.7 | 85.4 | 84.3 | 77.5 | 40.4 | 59.9 | 46.7 | 94.5 | 97.2 | 95.7 | 76.5 | 86.0 | 81.0 |
| | GRPO | 81.0 | 90.6 | 73.7 | 61.4 | 74.2 | 45.1 | 89.1 | 90.3 | 83.3 | 88.2 | 91.4 | 83.9 | 53.2 | 74.3 | 59.9 | 95.3 | 98.1 | 97.0 | 85.2 | 91.4 | 86.0 |
| | UniBias | 76.5 | 85.6 | 71.1 | 56.2 | 64.0 | 32.5 | 87.2 | 91.1 | 84.1 | 85.7 | 85.9 | 78.0 | 46.9 | 68.9 | 54.5 | 94.5 | 95.8 | 95.1 | 81.0 | 89.3 | 80.0 |
| | PriDe | 77.2 | 86.1 | 72.1 | 56.8 | 63.5 | 33.1 | 85.1 | 86.7 | 81.2 | 86.9 | 88.2 | 80.3 | 49.7 | 73.1 | 58.5 | 95.5 | 97.4 | 95.8 | 79.9 | 84.1 | 84.0 |
| | CalibraEval | 78.8 | 79.1 | 71.2 | 52.9 | 61.0 | 28.9 | 83.7 | 87.2 | 79.9 | 85.5 | 86.7 | 79.6 | 48.0 | 72.2 | 55.2 | 93.9 | 96.8 | 96.0 | 80.3 | 83.3 | 85.0 |
| | PIF | 79.4 | 86.5 | 72.8 | 62.2 | 68.4 | 36.9 | 89.3 | 90.6 | 85.5 | 90.8 | 90.9 | 84.7 | 52.1 | 72.8 | 60.1 | 96.1 | 97.6 | 96.8 | 84.3 | 92.0 | 88.0 |
| | PA-GRPO | 81.4 | 91.6 | 74.8 | 59.4 | 75.2 | 43.4 | 90.2 | 92.1 | 86.2 | 90.5 | 92.1 | 86.6 | 54.1 | 75.1 | 61.2 | 96.0 | 98.3 | 97.2 | 86.8 | 91.8 | 89.0 |
| | | (+0.4) | (+1.0) | (+1.1) | (-2.8) | (+1.0) | (-1.7) | (+0.9) | (+1.0) | (+0.7) | (-0.3) | (+0.7) | (+1.9) | (+0.9) | (+0.8) | (+1.1) | (-0.1) | (+0.2) | (+0.2) | (+1.6) | (-0.2) | (+1.0) |

Table 1: Main results on Llama-3.1-8B-Instruct, Qwen3-8B, and Qwen3-32B across 7 benchmarks. All scores are rounded to one decimal place. PA-GRPO results show the absolute score (top) and the gain/drop compared to the best baseline (bottom). **Bold** indicates the best result in each column.

| Method | LLM-as-a-Judge | | | | | | | | | Multiple Choice Questions | | | | | | | | | | | |
|-------------|----------------|-------------|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|---------------------------|-------------|-------------|-------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| | MT-Bench | | | JudgeBench | | | PreferenceBench | | | RewardBench | | | GPQA | | | ARC-Challenge | | | TinyMMLU | | |
| | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA |
| Base | 59.6 | 25.2 | 22.2 | 35.0 | 34.8 | 6.1 | 60.8 | 22.6 | 22.1 | 60.5 | 31.5 | 26.2 | 27.3 | 40.0 | 27.2 | 73.0 | 82.5 | 78.9 | 54.2 | 68.9 | 57.0 |
| GRPO | 75.7 | 80.6 | 65.4 | 48.2 | 56.1 | 28.2 | 82.2 | 85.1 | 76.3 | 70.9 | 76.9 | 61.5 | 29.5 | 55.6 | 29.7 | 89.0 | 92.1 | 91.5 | 67.3 | 77.9 | 73.0 |
| + r_{con} | 75.2 | 83.1 | 66.5 | 48.0 | 56.5 | 29.2 | 82.6 | 85.9 | 76.9 | 70.6 | 79.0 | 61.9 | 29.9 | 56.4 | 31.0 | 87.8 | 92.3 | 92.0 | 67.1 | 78.8 | 73.0 |
| + A_{PA} | 76.6 | 84.2 | 67.8 | 49.3 | 55.4 | 30.5 | 83.4 | 86.4 | 77.8 | 69.8 | 78.7 | 61.1 | 30.7 | 57.1 | 32.2 | 89.7 | 92.3 | 91.9 | 67.5 | 79.4 | 74.0 |
| PA-GRPO | 77.6 | 88.0 | 71.7 | 57.1 | 58.3 | 32.4 | 86.2 | 87.2 | 79.8 | 71.0 | 82.7 | 62.3 | 32.1 | 57.2 | 33.2 | 89.3 | 92.9 | 92.6 | 67.9 | 81.9 | 75.0 |

Table 2: Ablation study on Llama-3.1-8B-Instruct across all 7 benchmarks. We group datasets into LLM-as-a-Judge tasks and MCQ tasks. The best results are **bolded**. Adding both components (A_{PA} and r_{con}) consistently yields the best robustness.

stability under reordering. This aligns with our motivation that selection bias often manifests as unstable preferences across equivalent permutations.

For **Qwen3-8B**, the gain becomes more salient in LLM-as-a-Judge evaluations; for instance, JudgeBench accuracy reaches 60.1%, indicating that PA-GRPO regularizes the model more effectively in settings susceptible to selection bias, preventing it from relying on arbitrary presentation factors (e.g., position or label) for pairwise judgments. Meanwhile, the most notable improvement on MCQ tasks is observed on GPQA, where Consistent Accuracy improves from 46.6% to 56.7%, implying that the permutation-group consensus converges more reliably to the ground truth after filtering out permutation noise in complex scientific reasoning.

For the larger LLM, **Qwen3-32B**, although the absolute performance is already high, the gains remain persistent; MT-Bench consistency increases to 91.6%, suggesting that PA-GRPO continues to refine stability even with limited headroom. While a few benchmark-specific metrics show slight regressions, these appear isolated and do not alter the overall trend that enforcing permutation-aware optimization enhances stability and reliability across permutations in both LLM-as-a-Judge and multiple-choice evaluations. A qualitative case study illustrating how PA-GRPO tracks semantic content across permutations, in contrast to vanilla GRPO’s surface-level preference, is provided in Appendix G.

| Setting | LLM-as-a-Judge | | | | | | | | | | | | Multiple Choice Questions | | | | | | | | |
|---------------------------|----------------|-------------|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|-------------|-------------|---------------------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| | MT-Bench | | | JudgeBench | | | PreferenceBench | | | RewardBench | | | GPQA | | | ARC-Challenge | | | TinyMMLU | | |
| | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA |
| Base | 59.6 | 25.2 | 22.2 | 35.0 | 34.8 | 6.1 | 60.8 | 22.6 | 22.1 | 60.5 | 31.5 | 26.2 | 27.3 | 40.0 | 27.2 | 73.0 | 82.5 | 78.9 | 54.2 | 68.9 | 57.0 |
| $\lambda = 0.5$ | 78.0 | 86.4 | 69.8 | 56.5 | 55.4 | 31.9 | 85.3 | 86.8 | 78.9 | 70.7 | 80.9 | 60.5 | 32.1 | 56.6 | 32.0 | 89.4 | 92.0 | 92.4 | 67.1 | 78.2 | 74.0 |
| $\lambda = 1.0$ (Default) | 77.6 | 88.0 | 71.7 | 57.1 | 58.3 | 32.4 | 86.2 | 87.2 | 79.8 | 71.0 | 82.7 | 62.3 | 32.1 | 57.2 | 33.2 | 89.3 | 92.9 | 92.6 | 67.9 | 81.9 | 75.0 |
| $\lambda = 2.0$ | 76.5 | 88.6 | 70.5 | 56.8 | 55.9 | 32.0 | 85.0 | 87.5 | 79.1 | 70.3 | 82.5 | 62.1 | 31.4 | 58.0 | 33.6 | 88.7 | 91.2 | 91.1 | 66.4 | 80.1 | 73.0 |

Table 3: Impact of the consistency reward coefficient (λ) on Llama-3.1-8B-Instruct. We group datasets into LLM-as-a-Judge tasks and MCQ tasks. $\lambda = 1.0$ achieves the best balance. The best results are **bolded**.

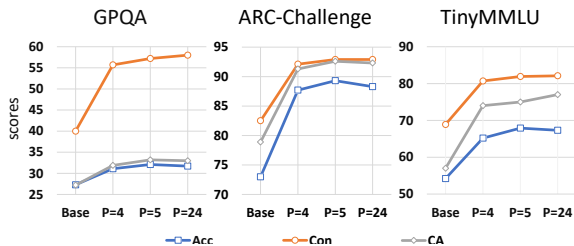


Figure 3: Impact of Permutation Group Size (P).

5.2 Ablation Study

Table 2 examines the contributions of the Consistency-Aware Reward r_{con} and the Cross-Permutation Advantage A_{PA} on Llama-3.1-8B-Instruct. Adding r_{con} to GRPO explicitly encourages within-group agreement, effectively raising MT-Bench consistency from 80.6% to 83.1% without sacrificing correctness. In contrast, enabling A_{PA} reduces permutation-induced reward variance during advantage estimation, which translates into broader gains, such as improving JudgeBench accuracy from 48.2% to 49.3%.

Crucially, the two components are complementary: r_{con} shapes the model toward agreement, while A_{PA} stabilizes the group-level optimization signal. Their combination yields the strongest overall robustness, boosting PreferenceBench consistent accuracy from 76.3% to 79.8% and reaching 75.0% on TinyMMLU. While minor benchmark-specific accuracy trade-offs are observed, the results confirm that both components are essential for reliable cross-permutation behavior.

5.3 Hyperparameter Analysis

We analyze two key hyperparameters of PA-GRPO on Llama-3.1-8B-Instruct: the consistency reward coefficient λ and the permutation group size P . Table 3 summarizes the effect of λ across LLM-as-a-Judge and MCQ benchmarks, and Figure 3 reports results for different P on MCQ tasks.

Consistency reward coefficient λ . We evaluate λ values of 0.5, 1.0, and 2.0 on Llama-3.1-8B-Instruct. We select $\lambda = 1.0$ as it optimally balances performance and consistency, achieving 79.8% consistent accuracy on PreferenceBench. In contrast, $\lambda = 0.5$ sacrifices stability for marginal accuracy gains on MT-Bench, while $\lambda = 2.0$ provides no significant performance benefit to justify the stronger regularization.

Permutation group size P . As illustrated in Figure 3, for MCQ tasks, we compare P values of 4, 5, and 24. We find that $P = 5$, which adds a reverse order to cyclic shifts, outperforms $P = 4$ by effectively disrupting adjacency bias. Although expanding to the full set where $P = 24$ increases TinyMMLU consistent accuracy from 75.0% to 77.0%, the marginal gain does not justify the nearly five-fold computational cost. We therefore adopt $P = 5$ for efficiency.

5.4 Chain-of-Thought Analysis

We further investigate whether the robustness gains of PA-GRPO depend on explicit chain-of-thought (CoT) generation. Table 4 compares four settings on Llama-3.1-8B-Instruct: Base (Direct), Base (CoT), PA-GRPO (Direct), and PA-GRPO (CoT).

The results reveal two complementary effects. First, PA-GRPO improves robustness even under direct decoding, indicating that permutation-aware alignment is internalized into the policy itself. For example, on MT-Bench, consistent accuracy improves from 58.0% for Base (CoT) to 69.3% for PA-GRPO (Direct).

Second, explicit reasoning further amplifies these gains. PA-GRPO (CoT) achieves the strongest overall results across most benchmarks, including GPQA, where consistent accuracy rises from 30.8% under direct decoding to 33.2% with CoT. This pattern suggests that CoT and permutation-aware training play different but com-

| Method | LLM-as-a-Judge | | | | | | | | | | | | Multiple Choice Questions | | | | | | | | |
|------------------|----------------|-------------|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|-------------|-------------|---------------------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| | MT-Bench | | | JudgeBench | | | PreferenceBench | | | RewardBench | | | GPQA | | | ARC-Challenge | | | TinyMMLU | | |
| | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA |
| Base (Direct) | 59.6 | 25.2 | 22.2 | 35.0 | 34.8 | 6.1 | 60.8 | 22.6 | 22.1 | 60.5 | 31.5 | 26.2 | 27.3 | 40.0 | 27.2 | 73.0 | 82.5 | 78.9 | 54.2 | 68.9 | 57.0 |
| Base (CoT) | 68.7 | 68.2 | 58.0 | 39.0 | 48.5 | 17.1 | 70.3 | 61.3 | 58.1 | 67.5 | 76.2 | 54.8 | 24.5 | 53.9 | 26.6 | 84.6 | 84.2 | 81.1 | 62.4 | 74.3 | 64.0 |
| PA-GRPO (Direct) | 75.1 | 85.5 | 69.3 | 58.0 | 54.5 | 31.8 | 84.8 | 82.3 | 75.9 | 69.5 | 80.1 | 59.8 | 31.0 | 54.1 | 30.8 | 86.5 | 89.9 | 88.7 | 66.5 | 77.9 | 72.0 |
| PA-GRPO (CoT) | 77.6 | 88.0 | 71.7 | 57.1 | 58.3 | 32.4 | 86.2 | 87.2 | 79.8 | 71.0 | 82.7 | 62.3 | 32.1 | 57.2 | 33.2 | 89.3 | 92.9 | 92.6 | 67.9 | 81.9 | 75.0 |

Table 4: Performance comparison of Direct vs. Chain-of-Thought (CoT) inference strategies. We report Accuracy, Consistency, and Consistent Accuracy for both the Base model and PA-GRPO across seven benchmarks.

| Bias Setting | LLM-as-a-Judge | | | | | | | | | | | | Multiple Choice Questions | | | | | | | | |
|----------------------------|----------------|-------------|-------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|-------------|-------------|---------------------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| | MT-Bench | | | JudgeBench | | | PreferenceBench | | | RewardBench | | | GPQA | | | ARC-Challenge | | | TinyMMLU | | |
| | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA | Acc | Con | CA |
| Label Bias (Label-only) | 76.7 | 92.2 | 72.8 | 56.3 | 79.0 | 45.8 | 74.9 | 62.8 | 56.2 | 70.7 | 87.5 | 64.4 | 22.0 | 67.2 | 20.8 | 80.7 | 95.6 | 81.3 | 63.5 | 83.8 | 66.0 |
| Position Bias (Order-only) | 77.4 | 85.1 | 69.9 | 55.0 | 45.5 | 27.7 | 74.2 | 66.9 | 57.7 | 71.6 | 82.0 | 62.6 | 27.9 | 65.0 | 27.7 | 80.9 | 91.9 | 82.8 | 64.3 | 80.5 | 72.0 |
| Selection Bias (Standard) | 77.6 | 88.0 | 71.7 | 57.1 | 58.3 | 32.4 | 86.2 | 87.2 | 79.8 | 71.0 | 82.7 | 62.3 | 32.1 | 57.2 | 33.2 | 89.3 | 92.9 | 92.6 | 67.9 | 81.9 | 75.0 |

Table 5: Bias decomposition results under three controlled protocols: Label Bias, Position Bias, and Selection Bias. We report Accuracy, Consistency, and Consistent Accuracy for PA-GRPO across seven benchmarks.

plementary roles: PA-GRPO improves the underlying policy’s invariance to label and position changes, while CoT provides additional deliberation that helps the model resolve difficult cases more consistently.

Overall, these results indicate that PA-GRPO does not depend on CoT to achieve robustness, but can benefit further from CoT at inference time when stronger reasoning is available.

5.5 Bias Decomposition Analysis

To better understand residual selection bias after PA-GRPO, we decompose it into three factors: label bias, position bias, and their coupled effect. We evaluate three protocols: **Label Bias**, which rotates option identifiers while preserving the content-position mapping; **Position Bias**, which permutes display order while preserving the identifier-content mapping; and **Selection Bias**, which applies the coupled identifier-position permutation used in our standard evaluation.

As shown in Table 5, residual sensitivity to position is often stronger than sensitivity to labels alone, especially on judge-style benchmarks. This trend is clearest on JudgeBench, where consistency is 79.0% under label-only perturbations but drops to 45.5% under order-only perturbations. In contrast, MCQ benchmarks generally exhibit weaker label effects; for example, ARC-Challenge reaches 95.6% consistency under label-only perturbations. These results suggest that residual bias is dataset-dependent and cannot be explained by a single fac-

tor.

We further find that the coupled protocol is most aligned with the training objective and is often competitive, especially on PreferenceBench. On PreferenceBench, the standard coupled protocol yields 87.2% consistency, compared with 62.8% under label-only perturbations and 66.9% under order-only perturbations. This is consistent with the design of PA-GRPO, which optimizes consistency under coupled permutations, although the strength of this advantage varies across datasets. The factor-isolated protocols remain informative because they reveal residual sensitivities that are not explicitly disentangled during training. Additional metrics (RStd, CKLD) and a prompt sensitivity analysis are reported in Appendix H.

6 Conclusion

In this paper, we proposed PA-GRPO, a novel RL framework designed to mitigate selection bias in LLMs. By integrating a cross-permutation advantage and a consistency-aware reward, our PA-GRPO explicitly enforces semantic invariance across different option arrangements. Extensive experiments on seven benchmarks demonstrate that PA-GRPO significantly outperforms strong baselines, effectively reducing position and label biases while maintaining high reasoning accuracy. This work highlights the efficacy of internalizing consistency constraints to achieve robust and reliable discrete-choice reasoning.

Limitations

While PA-GRPO proves effective, we acknowledge specific limitations in our current study scope. First, our method is primarily tailored for discrete-choice tasks (*e.g.*, MCQ and pairwise judgments) where permutation groups are naturally defined. Extending this consistency-based objective to open-ended generation remains a challenge, as semantic equivalence is harder to quantify in long-form text. Second, our evaluation is currently centered on English benchmarks and open-weights models. The interaction between permutation invariance and other systematic artifacts, such as verbosity bias (Dubois et al., 2024) or culture-specific biases in multilingual settings, has not yet been fully explored. We view our work as a foundational step toward permutation-robust alignment, leaving these broader generalization tests for future research.

Ethics Considerations

The research presented in this paper is founded on open-source LLMs and utilizes publicly available datasets. Consequently, we do not anticipate that our study will have any direct adverse effects. However, it is crucial to recognize that any generative AI technology, including the contributions of our research, must be implemented with caution to avert potentially harmful outcomes.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (72204087), the Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission (23CGA28), the Shanghai Pujiang Program (23PJC030), Young Elite Scientists Sponsorship Program by CAST (YESS20240562). We also appreciate the constructive comments from the anonymous reviewers.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An

open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

- 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.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- 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. *arXiv preprint arXiv:1803.05457*.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpaca-eval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- 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*.
- Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn’t always right. *arXiv preprint arXiv:2104.08315*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Joonwon Jang, Sanghwan Jang, Wonbin Kweon, Minjin Jeon, and Hwanjo Yu. 2024. Rectifying demonstration shortcut in in-context learning. *arXiv preprint arXiv:2403.09488*.
- Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2025. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797.
- Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhat-tacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2025a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791.
- Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. 2025b. 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.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.
- Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. *Advances in Neural Information Processing Systems*, 37:134387–134429.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- 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.
- Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Yuval Reif and Roy Schwartz. 2024. Beyond performance: Quantifying and mitigating label bias in llms. *arXiv preprint arXiv:2405.02743*.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.
- Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic study of position bias in llm-as-a-judge. *arXiv preprint arXiv:2406.07791*.
- Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *arXiv preprint arXiv:2410.12784*.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and 1 others. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450.

- Xiaoyue Wang and Xin Liu. 2025. Beyond generation: Leveraging llm creativity to overcome label bias in classification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25500–25506.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Yifei Wang, Feng Xiong, Yong Wang, Linjing Li, Xiangxiang Chu, and Daniel Dajun Zeng. 2025. Position bias mitigates position bias: Mitigate position bias through inter-position knowledge distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1495–1512.
- Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M Kakade, Hao Peng, and Heng Ji. 2024b. Eliminating position bias of language models: A mechanistic approach. *arXiv preprint arXiv:2407.01100*.
- 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, Shuang 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.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Soyoung Yoon, Dongha Ahn, Youngwon Lee, Minkyu Jung, HyungJoo Jang, and Seung-won Hwang. 2025. Rotor: Towards more reliable responses for order-invariant inputs. *arXiv preprint arXiv:2502.08662*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.
- Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.
- Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023a. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.
- Hanzhang Zhou, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. 2024. Unibias: Unveiling and mitigating llm bias through internal attention and ffnet manipulation. *Advances in Neural Information Processing Systems*, 37:102173–102196.

A Statement on the Use of Large Language Models

In this study, Large Language Models (LLMs) were employed solely for the purpose of grammatical correction and text refinement to improve readability. They were not involved in the conceptualization of the research, generation of content, or any substantive scientific contribution. The authors retain full responsibility for the content of this manuscript, ensuring its originality and accuracy.

B Training Procedure

Algorithm 1 Permutation-Aware GRPO (PA-GRPO)

Require: Policy π_θ , old policy π_{old} , reference policy π_{ref} , permutation set Π , parameters P, N, λ, β

```
1: for each batch  $\mathcal{B}$  do
2:   Initialize  $\mathcal{L}_{\text{batch}} \leftarrow 0$ 
3:   for each instance  $x \in \mathcal{B}$  do
4:     Construct  $\mathcal{G}(x) = \{p^{(t)} = \tau_t(x)\}_{t=1}^P$ 
5:     Sample  $N$  outputs  $y^{(t,i)}$  for each  $p^{(t)}$  and
       map them to semantic outputs  $z^{(t,i)}$ 
6:     Compute preprocessing rewards  $r_{\text{pre}}^{(t,i)}$ 
7:     if JUDGE then
8:       Compute consistency rewards  $r_{\text{con}}^{(t,i)}$  by
         pairwise agreement
9:     else if MCQ then
10:      Compute consistency rewards  $r_{\text{con}}^{(t,i)}$  by
        majority agreement
11:    end if
12:    Compute total rewards  $r^{(t,i)} = r_{\text{pre}}^{(t,i)} + \lambda r_{\text{con}}^{(t,i)}$ 
13:    Compute advantages  $A_{\text{PA}}^{(t,i)}$  using Eq. 12
14:    Compute instance objective  $\mathcal{L}_x$  using
       Eq. 15
15:     $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}_x$ 
16:  end for
17:  Update  $\theta$  using  $\mathcal{L}_{\text{batch}}$ 
18:  Set  $\pi_{\text{old}} \leftarrow \pi_\theta$ 
19: end for
```

C Benchmark Descriptions

This appendix briefly describes the seven evaluation benchmarks used in Section 4.1. We group them into (i) pairwise judging benchmarks, where each instance contains an instruction (or question) and two candidate responses with a reference preference label, and (ii) MCQ benchmarks, where

each instance is a 4-option multiple-choice question with a provided answer key. Unless otherwise noted, our permutation-based evaluation protocol (Appendix D) is applied to the canonical input format of each benchmark.

MT-Bench. MT-Bench (Zheng et al., 2023b) is a controlled, multi-turn instruction-following benchmark designed to evaluate conversational assistants. It consists of 80 manually written, high-quality multi-turn prompts spanning eight common user categories (*e.g.*, writing, roleplay, extraction, reasoning, math, coding, and two knowledge-oriented categories), with 10 prompts per category. Each MT-Bench prompt contains two turns, enabling evaluation of context tracking and multi-step dialogue behaviors. For pairwise judging settings, we use the standard pairwise formulation of MT-Bench prompts where a judge compares two model responses for the same prompt.

JudgeBench. JudgeBench (Tan et al., 2024) is a benchmark for evaluating the reliability of LLM-based judges under realistic pairwise comparison settings. Each example is a question paired with two candidate responses, accompanied by an objective preference label indicating which response is more correct. The released benchmark includes response pairs generated by strong frontier models, and is designed to probe common judging failure modes such as order sensitivity and bias.

PreferenceBench. PreferenceBench (Kim et al., 2024) is an in-domain pairwise ranking benchmark introduced alongside Prometheus 2. It is constructed by pairing two responses from a pool of multiple candidate responses per prompt, producing a set of 200 prompts with 2,000 labeled response pairs. In addition, PreferenceBench includes prompt-specific evaluation criteria intended to make the comparison more fine-grained and to stress judges' adherence to explicit rubrics.

RewardBench. RewardBench (Lambert et al., 2025) evaluates reward models (and judge-like scorers) on preference-style comparisons. Each instance is formatted as a prompt with a human-verified chosen completion and a rejected completion. The benchmark aggregates structured comparisons spanning several high-level categories, including chat instruction-following, reasoning-focused comparisons, and safety/refusal behavior, aiming to cover both in-distribution and challenging out-of-distribution preference cases.

ARC-Challenge. ARC (AI2 Reasoning Challenge) (Clark et al., 2018) is a multiple-choice science QA benchmark, where ARC-Challenge represents the more difficult subset. Questions are grade-school level science problems that typically require non-trivial reasoning beyond superficial retrieval. We use the standard ARC-Challenge configuration with its widely used train/validation/test splits and the provided answer key for evaluation.

TinyMMLU. TinyMMLU (Polo et al., 2024) is a compact evaluation set derived from MMLU, intended to approximate MMLU-style performance with far fewer questions. It consists of a small subsample of MMLU-format questions spanning diverse subject areas, each presented as a 4-option multiple-choice problem. This benchmark enables faster iteration while retaining broad topical coverage.

GPQA. GPQA (Rein et al., 2024) is a graduate-level, domain-expert-written multiple-choice QA benchmark designed to be difficult for non-experts even with access to web search. All questions are 4-option MCQ drawn from three high-level domains (biology, physics, and chemistry), with curated subsets that vary in strictness and difficulty (including a higher-quality, harder subset). We follow the standard GPQA evaluation protocol using the released answer key.

D Data Construction and Evaluation Details

Training Data Filtering. To ensure training efficiency and explicitly target selection bias, we performed pre-processing filtering on the source datasets (Chatbot Arena and MMLU). We utilized Llama-3.1-8B-Instruct to infer each training instance under both original and reversed orders (or cyclically shifted orders for MCQ). We specifically retained instances where the model yielded inconsistent predictions across permutations, as these samples provide the strongest signal for learning permutation invariance.

Full Permutation Protocol. During inference, we strictly evaluated the complete permutation space to calculate consistency metrics:

- **LLM-as-a-Judge Tasks:** For each pairwise comparison, we evaluated all $2! = 2$ possible orders: standard (AB) and reversed (BA).

- **MCQ Tasks:** For each 4-option question, we evaluated all $4! = 24$ possible arrangements of the options.

All reported metrics (Accuracy, Consistency, Consistent Accuracy) are aggregated results based on these fully expanded permutation groups.

E Baseline Details and Implementation

This appendix provides (i) the core idea of each baseline and (ii) how we implemented it in our experimental pipeline.

PriDe (Debiasing with Prior Estimation).

PriDe is a label-free, inference-time debiasing method originally proposed for multiple-choice selection robustness (Zheng et al., 2023a). Let q be the question, option-ID tokens be $\{d_i\}_{i=1}^n$ (e.g., A/B/C/D), and option contents be $\{o_i\}_{i=1}^n$. PriDe assumes the observed prediction distribution over option IDs can be decomposed into a prior term over d_i and a debiased term over the paired option content (under a permutation) (Zheng et al., 2023a):

$$P_{\text{obs}}(d_i | q, x^I) \propto P_{\text{prior}}(d_i | q) P_{\text{deb}}(o_{f_I(i)} | q, x), \quad (16)$$

where I indexes an option permutation (we use cyclic permutations as in the paper), and $f_I(i)$ maps the i -th ID token to the paired option content under I . With cyclic permutations \mathcal{I} , PriDe estimates a sample-specific prior by (Zheng et al., 2023a):

$$P_{\text{prior}}(d_i | q) = \text{softmax} \left(\frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \log P_{\text{obs}}(d_i | q, x^I) \right). \quad (17)$$

It then averages priors computed on a small estimation subset to obtain a global prior $\tilde{P}_{\text{prior}}(d_i)$, and debiases remaining samples by (Zheng et al., 2023a):

$$P_{\text{deb}}(o_i | q, x) \propto \frac{P_{\text{obs}}(d_i | q, x)}{\tilde{P}_{\text{prior}}(d_i)}. \quad (18)$$

For each dataset, we select a small subset of evaluation instances as the prior-estimation set. For each instance, we compute $P_{\text{obs}}(d_i | q, x^I)$ under cyclic permutations (for pairwise judging, $n=2$ reduces to swapping). We estimate per-instance

priors using the equation above, average them into a global prior \tilde{P}_{prior} , and then apply the debiasing ratio on the remaining instances using only the default prompt once (followed by renormalization and argmax).

CalibraEval (Label-free calibration for LLM-as-a-Judge). CalibraEval treats selection bias as a calibration problem and learns a mapping function $g(\cdot)$ that maps observed probabilities to an unbiased distribution (Li et al., 2025b):

$$P_{\text{deb}}(t_i | I, X_0) = g(P_{\text{obs}}(t_i | I, X_0)), \quad (19)$$

where t_i is an option-ID token (e.g., A/B) and X_0 is the default pairing of tokens and contents. It optimizes $g(\cdot)$ so that the calibrated judgments remain consistent under swapping positions and/or swapping ID tokens, and proposes a non-parametric order-preserving algorithm (NOA) to solve the resulting optimization (Li et al., 2025b).

We construct a small unlabeled estimation set and fit the calibration mapping under the corresponding swap or permutation variants required by the target task format. At test time, we apply the learned mapping to the observed probability distribution and renormalize it to obtain the final calibrated decision.

UniBias (Internal attention/FFN manipulation). UniBias identifies biased model components (attention heads and FFN vectors) by analyzing their contributions to label prediction, using three criteria: (i) relatedness to label prediction, (ii) bias/imbalance across labels, and (iii) low variance of the injected label information across samples; thresholds are selected via grid search (Zhou et al., 2024). After identifying biased components, UniBias mitigates bias by masking (eliminating) their influence during inference (Zhou et al., 2024).

We follow the UniBias procedure to build a small support set, identify biased attention heads and FFN vectors under the paper’s criteria and threshold selection, and apply the corresponding masks during evaluation. This keeps the inference-time complexity essentially identical to the original model once the masks are fixed (Zhou et al., 2024).

PIF (Point-wise Intelligent Feedback). PIF is an SFT algorithm designed to improve Multiple-Choice Symbol Binding (MCSB). It constructs negative instances by randomly combining incorrect option contents with all candidate symbols and

introduces a point-wise loss to provide direct feedback on these negatives, thereby reducing selection bias and improving accuracy (Xue et al., 2024).

We implement PIF under the same symbol-binding training principle in both task formats, constructing task-consistent point-wise negatives and optimizing the corresponding point-wise objective. We use the same backbone model and training budget as other training-side baselines for a fair comparison.

GRPO (Group Relative Policy Optimization).

GRPO is a PPO-style reinforcement learning algorithm that samples a group of outputs for the same prompt and uses relative (within-group) reward signals to optimize the policy, avoiding an explicit value model and reducing memory cost compared to standard PPO (Shao et al., 2024).

We implement GRPO as our direct RL baseline using the same reward design and training budget as PA-GRPO, but treat each permuted prompt as an independent sample and do not enforce cross-permutation consistency.

F Training Hyperparameters and Configurations

We implemented PA-GRPO based on the VeRL framework. Our experiments utilized three backbone models: **Llama-3.1-8B-Instruct**, **Qwen3-8B**, and **Qwen3-32B**. These models were evaluated across both MCQ and Judge tasks. To ensure computational efficiency, we leveraged FlashAttention (Dao et al., 2022) for accelerated attention computation and employed Low-Rank Adaptation (LoRA) on all linear layers with rank $r = 32$ and $\alpha = 64$. For Qwen3-8B and Qwen3-32B, we used the non-thinking mode to ensure a consistent decoding setting. The models were optimized using AdamW (Loshchilov and Hutter, 2017) with a learning rate of $1e-5$ for 2 epochs. We set the KL regularization (Jaques et al., 2019) coefficient $\beta = 0.001$ and the entropy coefficient to 0.001 to stabilize training. Rollout generation was accelerated using vLLM (Kwon et al., 2023) with a sampling size of $N = 8$ per prompt variant. Regarding batch configurations, we set the global batch size to 40 for MCQ tasks and 32 for Judge tasks to accommodate varying memory constraints. To ensure statistical reliability, all reported training results are averaged over 3 independent runs using different random seeds. For the evaluation phase, we set the temperature to 0 to strictly minimize sampling

noise and ensure the reproducibility of the selection bias metrics.

We additionally use a length regularization term and a format reward to regulate response length and structure, which helps stabilize outputs during training. The overall reward is jointly determined by the accuracy reward, the permutation-consistency reward, and these auxiliary rewards.

G Qualitative Case Study: Vanilla GRPO vs. PA-GRPO

To provide a concrete qualitative example, we present a case study from our evaluation that contrasts the behavior of vanilla GRPO and PA-GRPO under an answer-order permutation. The semantic contents of the two candidate answers are swapped between the original prompt and the permuted prompt, while the underlying task remains unchanged.

Task Setup. We consider a pairwise judgment example from the LLM-as-a-Judge evaluation. In the original prompt, the model is asked to compare two candidate answers to the following query: *“Thomas is very healthy, but he has to go to the hospital every day. What could be the reasons? Can you explain why the above question is interesting?”* In the permuted prompt, the semantic contents assigned to Answer A and Answer B are swapped. A permutation-consistent judge should therefore follow the stronger semantic content rather than preserving the same surface choice across the two prompt variants due to answer labels or positions.

Original Prompt

Given a question and two answers, determine which one better answers the question. Please think step by step. First, put your detailed reasoning and comparison inside `<think>` `</think>` tags. Then, provide your final decision inside `<answer>` `</answer>` tags. The answer should be a single letter (A or B).

Query:

Thomas is very healthy, but he has to go to the hospital every day. What could be the reasons? Can you explain why the above question is interesting?

Answer A:

There are several possible reasons why Thomas may have to go to the hospital daily even though he is healthy: ... The question is interesting for several reasons: ...

Answer B:

There could be several reasons for Thomas to visit the hospital daily despite being healthy. Some possible reasons include: ... The question is interesting for several reasons: ...

The permuted prompt is identical except that the semantic contents of Answer A and Answer B are swapped. This tests whether the model tracks semantic quality rather than answer labels or display position.

Vanilla GRPO. Vanilla GRPO produces the same surface choice under both prompt orders, selecting Answer A in both cases.

Original Response

`<think>`

Okay, let's tackle this. The user wants to know which answer (A or B) better explains why the question is interesting. The question is about Thomas being healthy but going to the hospital every day. The task is to compare the two answers and determine which one better addresses why the question is interesting.

First, I need to understand what each answer says. Answer A provides a list of possible reasons for Thomas's hospital visits and then explains why the question is interesting by listing six points. Answer B also lists possible reasons and then explains the interest in the question with five points.

Looking at the structure, both answers address the question's interest. But the user is asking which answer better explains why the question is interesting. So I need to compare the depth and breadth of their explanations.

Answer A's explanation of why the question is interesting includes points about paradox, lack of context, variety of explanations, psychological rewards, empathy, and open-endedness. These points are detailed and cover multiple angles: curiosity, deduction, emotional engagement, and the nature of the question itself.

Answer B's points are about challenging assumptions, empathy, diverse perspectives, critical thinking, and being a conversation starter. These are also valid, but maybe less comprehensive. Answer A's points are more varied and cover aspects like the psychological aspects (aha moment) and the paradox, which might make the question more intriguing.

Additionally, Answer A's explanation is more detailed, with six points, while Answer B has five. The depth of each point in Answer A seems more thorough. For example, Answer A mentions the paradox, lack of information, variety of explanations, emotional engagement, psychological reward, and open-endedness. Answer B's points are also good but perhaps not as comprehensive.

So, Answer A's explanation of why the question is interesting is more comprehensive and covers more angles. Therefore, Answer A better answers the question by providing a more detailed and varied explanation of the question's interest.

`</think>`

`<answer>A</answer>`

Permuted Response

<think>

Okay, let's tackle this. The user wants me to determine which answer (A or B) better addresses the original question. The question is about why Thomas, who's very healthy, has to go to the hospital every day. Then they ask why the question is interesting.

First, I need to compare the two answers. Let's look at Answer A. It lists several reasons like work, caregiving, volunteering, medical research, therapy, preventive care, and being a student. Then it explains why the question is interesting, pointing out assumptions, empathy, diverse perspectives, critical thinking, and conversation starters.

Answer B also lists possible reasons but includes more specific scenarios like visiting a sick family member, routine tests, clinical trials, working at the hospital, administrative tasks, blood donation, logistical reasons, and even questioning the accuracy of Thomas's health status. The explanation for why the question is interesting includes the paradox, lack of context, variety of explanations, psychological reward, empathy, and open-endedness.

Now, the original question is about the reasons for Thomas's daily hospital visits despite being healthy and why the question is interesting. Both answers cover the reasons, but I need to see which one is more comprehensive or better structured.

[... omitted several intermediate comparison steps for brevity ...]

Overall, both answers are good, but Answer B might be more detailed in the interest part. However, the user might prefer Answer A's structure.

In conclusion, both answers are good, but Answer A is more structured and covers the interest part with detailed points. Answer B has more specific scenarios.

I think I need to make a decision here. Based on the structure and the thoroughness, Answer A is better.

</think>

<answer>A</answer>

This behavior indicates that vanilla GRPO preserves the same surface choice under permutation, rather than consistently following the stronger semantic content after the answer contents are swapped.

PA-GRPO. PA-GRPO tracks the stronger semantic content across the two prompt variants, selecting Answer A on the original prompt and Answer B on the permuted prompt.

Original Response

<think>

Okay, let's tackle this. The user wants me to determine which answer (A or B) better explains why the question is interesting. The query is about Thomas being healthy but going to the hospital daily. The task is to compare

the two answers provided.

First, I need to understand the original question. It's a bit of a paradox because if someone is very healthy, why would they need to go to the hospital every day? The interesting part is the contradiction and the need to explore possible reasons beyond the obvious medical ones.

Now, looking at answer A. It lists eight possible reasons, ranging from caring for others, working at the hospital, participating in clinical trials, administrative tasks, blood donation, logistical reasons, and even the possibility that the health claim is incorrect. Then, it explains why the question is interesting by pointing out the paradox, the lack of context requiring speculation, the variety of explanations, the psychological reward of solving it, and the emotional engagement. The answer A seems thorough, covering a wide range of possibilities and explaining the question's interest in detail.

Answer B also lists several reasons: work, caregiving, volunteering, medical research, therapy, preventive care, and being a student. Then, it explains the question's interest by highlighting challenging assumptions, encouraging empathy, diverse perspectives, critical thinking, and being a conversation starter. Answer B is also comprehensive but perhaps a bit more concise in its explanations of why the question is interesting.

Now, comparing the two. Answer A's explanation of why the question is interesting is more detailed. It breaks down the reasons into six points, each with specific examples. Answer B's points are fewer and more general. For example, Answer A mentions the "paradoxical nature" and the "open-endedness" which are key aspects of the question's intrigue. Answer B talks about challenging assumptions and empathy, which are valid but perhaps less comprehensive.

Additionally, Answer A's list of possible reasons is more extensive, covering scenarios like administrative tasks and logistical reasons, which Answer B doesn't mention. However, Answer B's points about diverse perspectives and critical thinking are also valid. But the question specifically asks why the question is interesting, and Answer A's explanation is more thorough and covers more angles.

Another point: Answer A's explanation of the question's interest includes the psychological reward of solving a mystery, which adds depth. Answer B's points are good but perhaps not as detailed.

So, considering all that, Answer A provides a more comprehensive and detailed explanation of why the question is interesting, making it the better answer.

</think>

<answer>A</answer>

Permuted Response

<think>

Okay, let me try to work through this step by step. The user is asking which answer (A or B) better addresses the original question. The original question is about Thomas being very healthy but needing to go to the hospital every day. The user also wants to know why the

| Method / Setting | LLM-as-a-Judge | | | | | | | | Multiple Choice Questions | | | | | |
|---------------------------|----------------|--------------|--------------|--------------|-----------------|--------------|--------------|---------------|---------------------------|--------------|---------------|---------------|--------------|--------------|
| | MT-Bench | | JudgeBench | | PreferenceBench | | RewardBench | | GPQA | | ARC-Challenge | | TinyMMLU | |
| | RStd | CKLD | RStd | CKLD | RStd | CKLD | RStd | CKLD | RStd | CKLD | RStd | CKLD | RStd | CKLD |
| Base | 0.364 | 0.378 | 0.216 | 0.627 | 0.386 | 0.453 | 0.331 | 0.291 | 0.056 | 1.570 | 0.025 | 0.005 | 0.033 | 0.028 |
| GRPO | 0.036 | 0.010 | 0.060 | 0.156 | 0.057 | 0.009 | 0.024 | 0.006 | 0.042 | 1.331 | 0.019 | 0.001 | 0.026 | 0.030 |
| UniBias | 0.047 | 0.005 | 0.050 | 0.019 | 0.066 | 0.008 | 0.184 | 0.380 | 0.044 | 1.129 | 0.015 | 0.003 | 0.027 | 0.019 |
| PriDe | 0.078 | 0.012 | 0.107 | 0.023 | 0.051 | 0.006 | 0.029 | 0.002 | 0.036 | 0.983 | 0.014 | 0.002 | 0.029 | 0.031 |
| CalibraEval | 0.186 | 0.074 | 0.263 | 0.163 | 0.260 | 0.158 | 0.294 | 0.212 | 0.085 | 1.765 | 0.018 | 0.003 | 0.019 | 0.025 |
| PIF | 0.061 | 0.008 | 0.154 | 0.154 | 0.383 | 0.667 | 0.088 | 0.016 | 0.063 | 1.431 | 0.021 | 0.004 | 0.018 | 0.016 |
| PA-GRPO | 0.019 | 0.002 | 0.092 | 0.047 | 0.048 | 0.005 | 0.019 | 0.0007 | 0.012 | 0.662 | 0.009 | 0.0005 | 0.011 | 0.013 |
| PA-GRPO (Modified Prompt) | 0.025 | 0.003 | 0.086 | 0.051 | 0.055 | 0.005 | 0.016 | 0.0011 | 0.022 | 0.731 | 0.008 | 0.0007 | 0.014 | 0.011 |

Table 6: Bias evaluation results using RStd and CKLD (lower is better) on Llama-3.1-8B-Instruct across all seven benchmarks. All rows use the default evaluation prompt unless otherwise noted. The final row reports PA-GRPO under the rewritten prompt shown below. **Bold** indicates the best (lowest) result among all compared settings.

| | |
|---|---|
| <p>question is interesting.</p> <p>First, I need to compare the two answers provided. Both answers list several possible reasons for Thomas’s daily hospital visits. Let me check each part.</p> <p>Starting with the first part: the answers to the question. Answer A lists 7 reasons, including work, caregiving, volunteering, medical research, therapy, preventive care, and being a medical student. Answer B also lists 7 reasons, which include visiting a sick family member, routine tests, clinical trials, working at the hospital, administrative tasks, blood donation, logistical reasons, and the possibility that Thomas isn’t actually healthy.</p> <p>Now, looking at the second part: explaining why the question is interesting. Answer A mentions points like challenging assumptions, empathy, diverse perspectives, critical thinking, and conversation starters. Answer B talks about the paradox, lack of context, variety of explanations, the “aha” moment, empathy, and open-endedness.</p> <p>I need to see which answer better explains why the question is interesting. Both answers cover different aspects. Answer A focuses on the broader implications of the question, like challenging assumptions and promoting empathy. Answer B emphasizes the paradox and the process of deduction.</p> <p>Wait, looking at the original input: the user’s query is “Thomas is very healthy, but he has to go to the hospital every day. What could be the reasons? Can you explain why the above question is interesting?” Then the two answers (A and B) are responses to the entire query. So each answer includes both the list of reasons and the explanation of why the question is interesting.</p> <p>Looking at the first part (reasons): Answer A has 7 points, Answer B has 7 points. Both are comprehensive. However, Answer B includes an additional point about the possibility that Thomas isn’t actually healthy, which is a bit different.</p> <p>For the second part (why the question is interesting): Answer A lists 5 points, Answer B lists 6. Answer B’s explanation seems more detailed and covers more aspects.</p> | <p>So Answer B is more comprehensive in the first part.</p> <p>In the second part, Answer B’s explanation is more detailed.</p> <p>In conclusion, Answer B provides a more comprehensive and detailed explanation of both parts of the question, making it the better answer.</p> <p></think></p> <p><answer>B</answer></p> <p>This example qualitatively illustrates the behavioral difference between the two methods: vanilla GRPO may preserve a surface preference across permutations, whereas PA-GRPO more reliably follows the semantically stronger answer.</p> <h2>H Bias-Specific Evaluation Metrics and Prompt Sensitivity Analysis</h2> <p>To further examine whether the gains of PA-GRPO reflect reduced selection bias rather than overfitting to a specific prompt template, we report two additional bias-oriented metrics together with a prompt-variation experiment. All experiments in this section are conducted on Llama-3.1-8B-Instruct.</p> <p>In addition to the Accuracy, Consistency, and Consistent Accuracy metrics used in the main paper, we introduce two complementary metrics (lower is better) that directly quantify the degree of selection bias from a distributional perspective: (1) RStd (Recall Standard Deviation) (Zheng et al., 2023a), which measures the dispersion of per-choice recalls across answer IDs; a lower RStd indicates less uneven preference over answer labels; and (2) CKLD (Choice Kullback-Leibler Divergence) (Choi et al., 2025), which measures the KL divergence between the predicted choice-label distribution and the ground-truth label distribution; a lower CKLD indicates closer distributional align-</p> |
|---|---|

ment.

Table 6 reports the RStd and CKLD results across all seven benchmarks. The last row additionally reports PA-GRPO under a substantially rewritten evaluation prompt to test prompt sensitivity.

PA-GRPO achieves the lowest RStd and CKLD values on most benchmarks, indicating both reduced dispersion in per-choice recalls and better alignment between the predicted and ground-truth choice-label distributions. UniBias is slightly better on JudgeBench under these two metrics, but this advantage does not translate into better overall consistency in Table 1. This comparison suggests that explicitly suppressing a local bias mechanism is not always sufficient to preserve broader reasoning quality across datasets.

Default Prompt

Given a question and two answers, determine which one better answers the question. Please think step by step. First, put your detailed reasoning and comparison inside `<think>` `</think>` tags. Then, provide your final decision inside `<answer>` `</answer>` tags. The answer should be a single letter (A or B).

Modified Prompt

Please review a given question alongside two provided answers, and evaluate which response addresses the prompt more effectively. You must think through your decision step by step.

Evaluation Guidelines

Focus on Quality: Assess the answers based on their accuracy, relevance, completeness, and clarity.

Maintain Strict Objectivity: Do not let external factors influence your judgment. Remain unbiased with respect to the order in which the answers are presented and the labels assigned to them (A or B).

Begin by placing your detailed reasoning and comparative analysis within `<think>` `</think>` tags. Afterward, output your final verdict inside `<answer>` `</answer>` tags. The final answer must be exactly one letter: A or B.

Compared with the default-prompt row in Table 6, the modified-prompt row remains close across all seven benchmarks. This result suggests that the improvement is not tied to a particular surface template, but instead reflects a more stable reduction in selection bias.