

# DISCO Balances the Scales: Adaptive Domain- and Difficulty-Aware Reinforcement Learning on Imbalanced Data

Yuhang Zhou<sup>1\*</sup> Jing Zhu<sup>3\*</sup> Shengyi Qian<sup>3</sup> Zhuokai Zhao<sup>4</sup> Xiyao Wang<sup>1</sup>  
Xiaoyu Liu<sup>1</sup> Ming Li<sup>1</sup> Paiheng Xu<sup>1</sup> Wei Ai<sup>1</sup> Furong Huang<sup>1,2</sup>

<sup>1</sup> University of Maryland, College Park <sup>2</sup> Capital One

<sup>3</sup> University of Michigan, Ann Arbor <sup>4</sup> University of Chicago

{tonyzhou, xywang, xliu1231, paiheng, minglii, aiwei, furongh}@umd.edu  
{jingzhu, syqian}@umich.edu zhuokai@uchicago.edu

## Abstract

Large Language Models (LLMs) are increasingly aligned with human preferences through Reinforcement Learning from Human Feedback (RLHF). Among RLHF methods, Group Relative Policy Optimization (GRPO) has gained attention for its simplicity and strong performance, notably eliminating the need for a learned value function. However, GRPO implicitly assumes a balanced domain distribution and uniform semantic alignment across groups—assumptions that rarely hold in real-world datasets. When applied to multi-domain, imbalanced data, GRPO disproportionately optimizes for dominant domains, neglecting underrepresented ones and resulting in poor generalization and fairness. We propose **Domain-Informed Self-Consistency Policy Optimization (DISCO)**, a principled extension to GRPO that addresses inter-group imbalance with two key innovations. *Domain-aware reward scaling* counteracts frequency bias by reweighting optimization based on domain prevalence. *Difficulty-aware reward scaling* leverages prompt-level self-consistency to identify and prioritize uncertain prompts that offer greater learning value. Together, these strategies promote more equitable and effective policy learning across domains. Extensive experiments across multiple LLMs and skewed training distributions show that DISCO improves generalization, outperforms existing GRPO variants by 5% on Qwen3 models, and sets new state-of-the-art results on multi-domain alignment benchmarks. Our code and data are available at [https://github.com/Tonyzhou98/disco\\_grpo](https://github.com/Tonyzhou98/disco_grpo)

## 1 Introduction

Aligning large language models (LLMs) with human preferences is a central challenge in modern

AI systems (OpenAI, 2023; Touvron et al., 2023; Yang et al., 2024). Reinforcement Learning from Human Feedback (RLHF) has become the dominant approach for fine-tuning LLMs toward desirable behavior, enabling alignment with nuanced human intent (Ouyang et al., 2022; Shao et al., 2024; Zheng et al., 2023; Rafailov et al., 2023; Zhou et al., 2023; Wang et al., 2024, 2025). Within this framework, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) offers a promising alternative to value-based methods, simplifying training while achieving strong performance.

Despite its advantages, GRPO faces a significant challenge when applied to multi-domain datasets, a common scenario in real-world applications spanning math, question answering, and commonsense reasoning. While GRPO effectively removes the need for a value network and mitigates within-group variance, it implicitly assumes that prompt groups are sampled uniformly and that reward signals are semantically aligned across domains. However, this assumption often fails in practice. Real-world datasets are typically imbalanced, with a few dominant domains and many underrepresented ones (Zhou et al., 2024). Optimization gradients become skewed toward high-frequency domains, starving rare domains of learning signal. This results in models that generalize poorly to critical low-resource domains and amplify existing data biases. Data augmentation offers one workaround but introduces substantial overhead in generating high-quality synthetic prompts (Tepper et al., 2020).

To address the inter-group imbalance in GRPO, we propose **Domain-Informed Self-Consistency Policy Optimization (DISCO)**, an enhanced framework designed to promote equitable learning across imbalanced multi-domain datasets. DISCO introduces two complementary strategies to improve generalization under distributional skew: *domain-aware* and *difficulty-aware* reward scaling.

*Domain-aware scaling* reweights prompt groups

\*Equal contribution.

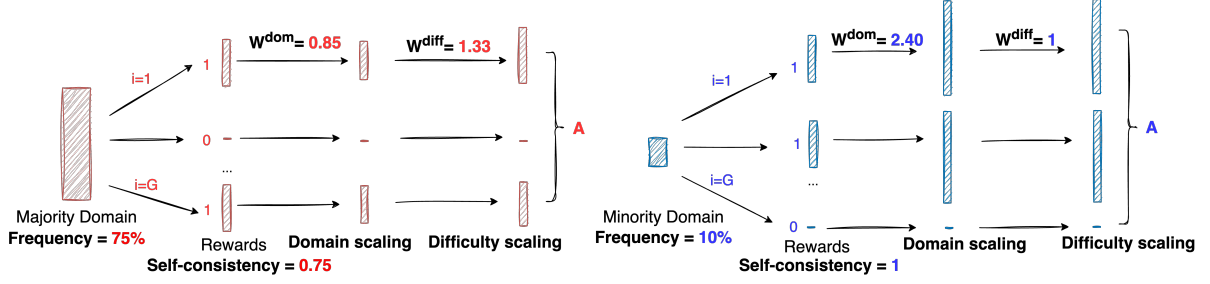


Figure 1: **Overview of the proposed DISCO scaling framework.** The framework is composed of two strategies to enhance GRPO’s robustness: (1) domain-aware scaling, which reweights prompt groups based on domain frequency, and (2) difficulty-aware scaling, which encourages the model to focus more on uncertain samples based on self-consistency scores.  $w^{\text{dom}}$  and  $w^{\text{diff}}$  denote the domain and difficulty weight.  $\underline{G}$ =Group size,  $\underline{A}$ =Advantage.

inversely by their frequency, reducing over-optimization on dominant domains while amplifying learning signals from underrepresented ones. *Difficulty-aware scaling* leverages prompt-level self-consistency, an intrinsic signal in GRPO, to identify and upweight prompts where the policy exhibits high uncertainty or inconsistent responses. Since not all prompts are equally challenging, treating them uniformly can lead the policy to overfit on easy examples while neglecting harder, more informative ones. By prioritizing uncertain prompts, this strategy guides the model to focus its learning on cases that offer a greater signal for improvement.

By integrating these two forms of adaptive scaling, DISCO enables all domains, regardless of their prevalence, to meaningfully contribute to policy optimization. As a result, it effectively mitigates GRPO’s inter-group imbalance and achieves state-of-the-art performance across diverse LLM architectures and training distributions.

Our contributions are summarized as follows:

- **Systematic Analysis of GRPO under data imbalance:** We perform the first systematic analysis on GRPO’s inherent vulnerability to dataset imbalance, empirically demonstrating its resulting performance degradation on underrepresented domains.
- **Strategic Framework:** Our novel DISCO framework introduces a powerful integration of domain-aware and difficulty-aware reward scaling strategies, effectively addressing the inter-group imbalance that limits standard GRPO.
- **SoTA Performance:** Our comprehensive empirical evaluations confirm that DISCO improves existing GRPO algorithms by 5% on Qwen3 models across diverse benchmarks and sets new standards for generalization performance.

## 2 Background and Motivation

This section first provides the necessary background on the GRPO algorithm. We then present an empirical study to concretely demonstrate how its optimization behavior is affected by domain imbalance in training data, thereby establishing the specific motivation for the enhancements proposed in subsequent sections.

### 2.1 Group Relative Policy Optimization

While standard RLHF relies on learned value networks, GRPO (Shao et al., 2024) computes relative advantages directly within each prompt group, simplifying training. Specifically, for a given prompt  $q$ ,  $G$  candidate outputs  $\{o_i\}_{i=1}^G$  are sampled from the old policy model  $\pi_{\theta_{\text{old}}}$ . A reward  $r_i$  is assigned to each  $o_i$ , and the group-normalized advantage  $A_i$  is calculated by normalizing the rewards across the group, which is:

$$A_i = \frac{r_i - \bar{r}}{\sigma_r} \quad (1)$$

where  $\bar{r} = \frac{1}{G} \sum_{j=1}^G r_j$  is the group mean reward, and  $\sigma_r$  is the standard deviation of the rewards within the group. The overall GRPO objective is given by:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\}} \left[ \frac{1}{G} \sum_{i=1}^G \min \left( \frac{\pi_{\theta}(o_i | q)}{\pi_{\theta_{\text{old}}}(o_i | q)} A_i, \text{clip} \left( \frac{\pi_{\theta}(o_i | q)}{\pi_{\theta_{\text{old}}}(o_i | q)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right) \right] - \beta \mathbb{D}_{\text{KL}}[\pi_{\theta} \| \pi_{\text{ref}}] \quad (2)$$

where  $\pi_{\theta}$  and  $\pi_{\theta_{\text{old}}}$  denote the new and old policies respectively,  $A_i$  is the group-normalized advantage, ‘clip’ denotes the clip function, and  $\mathbb{D}_{\text{KL}}$  is a KL regularization term encouraging the updated policy to remain close to a reference policy.

In this work, we focus on a rule-based GRPO setup, similar to DeepSeek’s R1 model train-

	IMDB	GSM8K	Math	NQ	ARC	Avg.
Balanced	89.60	57.94	23.82	14.90	47.44	<b>46.74</b>
Math heavy	87.80	<b>58.98</b>	<b>24.38</b>	13.96	45.99	46.22
IMDB heavy	<b>90.90</b>	56.88	23.54	13.85	45.90	46.21
NQ heavy	89.40	56.22	23.36	<b>16.57</b>	46.58	46.43
ARC heavy	88.80	55.29	22.46	14.16	<b>48.29</b>	45.80

Table 1: Performance (Exact Match (EM) Accuracy %) of Qwen2.5-0.5B trained with GRPO ( $G = 4$ ) under various domain-heavy training distributions.

ing (Guo et al., 2025), with deterministic rewards: 1 for an exact match (EM) with the ground-truth answer, 0 otherwise. This controlled setup allows us to isolate optimization and data distribution effects on learning dynamics, avoiding the variability introduced by learned reward models.

## 2.2 Empirical Demonstration of Domain Imbalance Impact

While GRPO normalizes advantages locally within prompt groups, its global optimization trajectory can be unduly influenced by the frequency of domains in the training data, a vulnerability already highlighted in Section 1. To empirically demonstrate this limitation, we performed an experiment training Qwen2.5-0.5B (Yang et al., 2024) with GRPO using datasets featuring distinct domain compositions and present the performance in Table 1. Specifically, we trained multiple models, each on 4,000 examples. For each model, the training data composition was intentionally skewed by heavily weighting one domain (e.g., 3,000 math prompts for a ‘Math-heavy’ model) while underrepresenting the others. Table 2 and Appendix A provide details on data and implementation.

As shown in Table 1, domain-heavy training biases GRPO performance: models excel in the over-represented domain but underperform on others compared to a balanced setup. For example, the Math-heavy model performs best on math tasks (24.38% Math, 58.98% GSM8K), but its average score across all domains is lower than the balanced model’s, primarily due to sub-optimal performance on other domains such as IMDB and ARC. This trade-off is consistent across other skewed settings, where specialization gains are accompanied by performance drops in underrepresented domains.

These results confirm that GRPO lacks a mechanism for inter-group calibration, and its optimization may become biased toward more frequently sampled domains, leading to suboptimal performance on rare but important tasks. This observa-

Dataset	Task Domain
IMDB (Maas et al., 2011)	Text Classification (TC)
GSM8K (Cobbe et al., 2021)	Mathematical Reasoning
MATH (Hendrycks et al., 2021)	Mathematical Reasoning
Natural Questions (NQ) (Lee et al., 2019)	Open-domain QA
ARC (Clark et al., 2018)	Multi-step Reasoning QA

(a) Test datasets and their corresponding domains. ARC denotes both the ARC-Easy and -Challenge subsets. All evaluations are conducted using exact match (EM).

Setup	Math	NQ	ARC	IMDB
Balanced	25%	25%	25%	25%
Math-heavy	75%	8.3%	8.3%	8.4%
NQ-heavy	8.3%	75%	8.3%	8.4%
ARC-heavy	8.3%	8.3%	75%	8.4%
IMDB-heavy	8.3%	8.3%	8.4%	75%

(b) Training Distribution by Domain (Proportions)

Table 2: Overview of evaluation and training datasets. ‘‘Heavy’’ settings allocate 75% of training prompts to a single domain, with the remaining 25% distributed equally among the others. For the math domain, we sample from the MetaMath dataset (Yu et al., 2023) for training, while for all other domains, we use the training portion of each corresponding evaluation benchmark.

tion motivates our investigation into domain-aware and difficulty-aware reward scaling methods to improve robustness in imbalanced settings.

## 3 Domain- and Difficulty-Aware Scaling

Building on the observation that GRPO lacks mechanisms for addressing domain-level imbalance, our DISCO, as depicted in Figure 1, proposes two modifications in multi-domain settings: **domain-aware** and **difficulty-aware reward scaling**.

First, we introduce domain-aware scaling, which reweights prompt groups based on domain frequency. This strengthens the learning signal for underrepresented ones.

Secondly, our difficulty-aware scaling enhances learning efficiency by focusing on challenging examples. It uses prompt-level self-consistency, the average reward from  $G$  candidate outputs per prompt, as a signal of policy uncertainty within GRPO. A lower self-consistency indicates higher uncertainty and thus greater difficulty. Group rewards are then scaled by a weight inversely proportional to this self-consistency score, thereby prioritizing more ambiguous prompts.

Together, these two mechanisms allow GRPO to adaptively rescale group-level advantages based on both domain rarity and prompt difficulty, offering a more globally informed optimization process while preserving its core group-relative framework.

### 3.1 Domain-Aware Scaling

Vanilla GRPO’s group-level normalization operates independently of a prompt group’s originating domain frequency. In imbalanced datasets (Section 2.2), this means high-frequency domains disproportionately influence the aggregated optimization gradient, potentially marginalizing low-frequency domains. To mitigate this issue, we introduce a **domain-aware reward scaling** strategy. For each prompt group  $q$  from domain  $d$ , we apply a domain weight  $w^{\text{dom}}(q)$  to rescale its rewards:

$$r_i^{\text{scaled}} = r_i \cdot w^{\text{dom}}(q), \quad (3)$$

and compute the group-level advantage as:

$$A_i = r_i^{\text{scaled}} - \bar{r}^{\text{scaled}}, \quad (4)$$

where  $\bar{r}^{\text{scaled}} = \frac{1}{G} \sum_{j=1}^G r_j^{\text{scaled}}$  is the group mean.

Note that we do not apply standard deviation normalization, as in the original GRPO (Equation 1), in order to preserve the absolute scaling effect of the domain weights. This design allows domain frequency to directly modulate the magnitude of the advantage signal, enabling rarer domains to exert a stronger influence during policy updates.

**Domain Weight Variants.** We explore three definitions of the domain weight based on domain frequency. Let  $p_d$  denote the proportion of prompt groups from domain  $d$ :  $p_d = \frac{N_d}{\sum_{d'} N_{d'}}$ , where  $N_d$  is the number of prompts from domain  $d$ .

We consider the following three scaling variants:

$$\mathbf{v1 \text{ (log)}}: \quad w^{\text{dom}} = \log \left( 1 + \frac{1}{p_d} \right) \quad (5)$$

$$\mathbf{v2 \text{ (log-squared)}}: \quad w^{\text{dom}} = \left[ \log \left( 1 + \frac{1}{p_d} \right) \right]^2 \quad (6)$$

$$\mathbf{v3 \text{ (inverse)}}: \quad w^{\text{dom}} = \frac{1}{p_d} \quad (7)$$

These variants are chosen to represent a spectrum of upweighting strengths. v1 (log) is hypothesized to provide a tempered yet significant boost to underrepresented domains; the logarithmic function naturally moderates the impact of extreme  $p_d$  values, potentially enhancing training stability. v2 (log-squared) represents a more assertive non-linear scaling. v3 (inverse) offers the most direct and aggressive form of upweighting, making domain weights sharply inversely proportional to their frequency. This systematic variation from a more

conservative (v1) to a highly aggressive (v3) approach allows us to study how different levels and types of domain correction affect training dynamics and model performance. A comparative analysis of these variants is presented in Section 5.1.

While domain-aware scaling helps rebalance across domains, it does not address intra-domain variation in prompt difficulty. Even within a single domain, some prompts are trivial while others are ambiguous. To account for this finer-grained challenge, we introduce difficulty-aware scaling.

### 3.2 Difficulty-Aware Scaling

While domain-aware weighting corrects for domain imbalance, it does not consider variation in prompt-level difficulty. As shown in empirical results (Section 2.2), domains like IMDB achieve consistently high accuracy, whereas others such as Math yield lower accuracy, indicating that prompt complexity varies widely even under balanced training.

The GRPO algorithm provides a natural mechanism for estimating prompt-level difficulty. For each prompt  $q$ , GRPO samples a group of  $G$  candidate completions  $\{o_i\}_{i=1}^G$ , each associated with a binary reward  $r_i \in \{0, 1\}$ . Previous work has shown that self-consistency, the agreement across model outputs, can serve as a proxy for uncertainty (Wang et al., 2022; Li et al., 2023). Prompt groups with mixed outcomes reflect uncertainty and may benefit from greater optimization focus.

To capture this, we define the self-consistency (SC) score for prompt  $q$  as:  $\text{SC}(q) = \frac{1}{G} \sum_{i=1}^G r_i$ , and define the difficulty weight as:

$$w^{\text{diff}}(q) = \frac{1}{\text{SC}(q) + \epsilon'}, \quad (8)$$

where  $\epsilon'$  is a small constant to ensure numerical stability.

Note that when all generations in a group are incorrect (i.e.,  $\text{SC}(q) = 0$ ),  $w^{\text{diff}}$  becomes large. However, this does not lead to instability, as all advantages will be zero based on Equation 4, and thus no policy update occurs. This mechanism encourages the model to focus more on prompts it finds uncertain, while ignoring uniformly poor or trivially easy cases.

Combining both components, we compute the final scaled reward of DISCO as:

$$r_i^{\text{scaled}} = r_i \cdot w^{\text{dom}}(q) \cdot w^{\text{diff}}(q), \quad (9)$$

where  $w^{\text{dom}}(q)$  and  $w^{\text{diff}}(q)$  are the domain- and difficulty-based weights for prompt group  $q$ , respec-



tively. The final advantage is then computed using Equation 4. This formulation retains the structure of GRPO while enhancing it with principled scaling to reflect domain imbalance and prompt difficulty.

## 4 Experiment Setup

We now evaluate the effectiveness of our DISCO. Our experiments aim to assess whether these methods improve alignment performance on underrepresented or difficult domains, and whether the improvements are consistent across different models.

**Dataset Setup.** We evaluate across four task domains: IMDB (text classification), GSM8K and MATH (math problem solving), NATURAL QUESTIONS (open-domain QA), and ARC (reasoning QA). Our training dataset consists of 4,000 examples, as detailed in Table 2.

**Baseline Methods.** We compare our method against the following baselines: (1) **Base Model**, the pretrained model without any fine-tuning; (2) **Naive GRPO**, which applies the original group-relative optimization without any reward reweighting; and (3) **Dr. GRPO** (Liu et al., 2025), which removes the length and standard deviation normalization. While other GRPO variants address issues like length bias or training instability (Yu et al.; Zhang et al., 2025b,a), they are not included as baselines as our work targets GRPO’s vulnerability to domain imbalance. In addition, we conduct ablation studies to isolate the contributions of domain-aware and difficulty-aware components.

**Proposed Methods.** We evaluate three variants of DISCO, each utilizing one of the domain-aware weighting strategies in Section 3.1. **DISCO-Log** uses log-scaled domain weights (v1), **DISCO-LogSq** uses squared-log weights (v2), and **DISCO-Inv** uses inverse-frequency weights (v3). All variants incorporate the same difficulty-aware component based on self-consistency. This setup allows us to assess how different strengths of domain correction interact with difficulty-aware scaling in multi-domain alignment.

**Model Setup.** We evaluate our method across a diverse set of language models. These include **Qwen2.5-0.5B**, **Qwen2.5-1.5B**, **Qwen2-0.5B**, **Qwen3-0.6B**, **Qwen2.5-7B**, and **Qwen1.5-MoE-A2.7B** (14B total parameters, 2.7B activated) (Yang et al., 2024), as well as **LLaMA3.2-1B** (Grattafiori et al., 2024), **Olmo2-1B** (OLMo et al.,

2024), and **Gemma2-2B-it** (Team, 2024). This selection spans both dense and mixture-of-experts (MoE) architectures and covers a range of model capacities from 0.5B to 14B parameters. For the GRPO group size, we use  $G = 2, 4, 8, 16$  depending on the experiment. Additional implementation details are provided in Appendix A.

## 5 Results

We begin our analysis by identifying the most effective variant of our proposed scaling strategy. After selecting a default method, we evaluate its impact through comparisons with baseline methods and targeted ablation studies.

### 5.1 Identifying the Optimal Scaling Strategy

To determine the most effective variant of DISCO, we compare three domain weighting strategies: **DISCO-Log** (v1), **DISCO-LogSq** (v2), and **DISCO-Inv** (v3), across several models trained under domain-heavy conditions (group size  $G = 4$ ). Table 3 summarizes these results. Each column shows the average accuracy across five evaluation datasets achieved when training under the specified domain-heavy condition. The final column (‘Avg.’) averages these scores across the four conditions.

DISCO-Log achieves the highest average accuracy (‘Avg.’ column) across all tested models, striking the best balance between performance improvement and stability. While the more aggressive DISCO-Inv or DISCO-LogSq occasionally outperforms in specific settings (e.g., IMDB-heavy for LLaMA3.2-1B using v3), this often comes at the cost of reduced performance elsewhere, lowering their overall effectiveness. In contrast, DISCO-Log delivers the top overall average score for all three models in this comparison. When averaged across all domain-model combinations shown in Table 3, DISCO-Log surpasses DISCO-LogSq by 1.27% and DISCO-Inv by 0.97%.

DISCO-Log’s superior performance, given that minority domains constitute only 8.3% of our data, strongly aligns with logarithmic scaling’s theoretical benefits. Its inherent diminishing returns temper responses to extreme domain characteristics, preventing over-correction and fostering a stable learning signal. In contrast, the aggressive scaling of DISCO-LogSq and DISCO-Inv, while occasionally beneficial in specific scenarios, led to higher performance variance and risked broader instability due to this pronounced data imbalance, thus

Model	Math-heavy	IMDB-heavy	NQ-heavy	ARC-heavy	Avg.
<i>Qwen2.5-0.5B</i>					
DISCO-Log	<b>47.92</b>	47.67	<b>47.91</b>	<b>47.70</b>	<b>47.80</b>
DISCO-LogSq	47.75	47.47	47.36	45.96	47.14
DISCO-Inv	47.72	<b>47.81</b>	47.59	46.22	47.33
<i>Qwen2.5-1.5B</i>					
DISCO-Log	64.12	64.93	<b>65.26</b>	<b>65.35</b>	<b>64.91</b>
DISCO-LogSq	<b>65.33</b>	<b>65.21</b>	64.06	64.56	64.79
DISCO-Inv	65.26	64.47	64.67	64.92	64.83
<i>LLaMA3.2-1B</i>					
DISCO-Log	<b>28.97</b>	30.02	22.19	<b>30.56</b>	<b>27.94</b>
DISCO-LogSq	27.49	25.51	22.46	24.22	24.92
DISCO-Inv	22.57	<b>30.10</b>	<b>25.34</b>	24.27	25.58

Table 3: Average accuracy under each domain-heavy training setup for different reward scaling variants. Bold denotes the highest score for each condition.

explaining their lower average scores. Prioritizing robust alignment across domains through consistent and strong average performance, we therefore adopt DISCO-Log for all subsequent experiments.

## 5.2 Comparison with Baselines

**Our method outperforms baselines across different models and training distributions.** To evaluate the effectiveness of DISCO, we compare DISCO against two key baselines: Naive GRPO, the original formulation without reward rescaling, and Dr. GRPO (Liu et al., 2025). These comparisons utilize models trained ( $G = 4$ ) under the various domain-heavy distributions. We summarize the comparison in Table 4, which shows the average performance across all datasets under each setting. Complete results for individual datasets and additional base models, are shown in Appendix C, Table 8 and Table 9.

Focusing on Table 4, we first note that all GRPO variants evaluated consistently yield substantial improvements over their respective base models, confirming the effectiveness of GRPO-based alignment (Guo et al., 2025). Turning to the comparison between the GRPO variants, we observe from the overall average performance (‘Avg.’ column) that **DISCO achieves the highest overall average score on 5 out of the 6 models**. The most significant gains appear on the Qwen3-0.6B model, where DISCO improves upon Naive GRPO by 4.40% and upon Dr. GRPO by 4.61%. Strong gains are also seen on LLaMA3.2-1B (+1.65% vs Naive, +1.97% vs Dr. GRPO) and Qwen2.5-0.5B (+1.63% vs Naive, +0.65% vs Dr. GRPO). Furthermore, DISCO delivers the best average on the Qwen1.5-MoE (+0.39% vs Naive, +1.44% vs Dr. GRPO). On Qwen2.5-1.5B, performance is very close, with Naive GRPO slightly ahead in

Model	Math-heavy	IMDB-heavy	NQ-heavy	ARC-heavy	Avg.
<i>Qwen2.5-0.5B</i>					
Base	42.80				
Naive GRPO	46.22	46.21	46.43	45.80	46.17
Dr GRPO	46.90	46.22	47.84	47.62	47.14
DISCO	<b>47.92</b>	<b>47.67</b>	<b>47.91</b>	<b>47.70</b>	<b>47.80</b>
<i>Qwen2.5-1.5B</i>					
Base	60.06				
Naive GRPO	<b>64.81</b>	64.91	65.15	65.02	<b>64.97</b>
Dr GRPO	64.76	64.75	64.72	65.16	64.85
DISCO	64.12	<b>64.93</b>	<b>65.26</b>	<b>65.35</b>	64.91
<i>Qwen2-0.5B</i>					
Base	43.18				
Naive GRPO	44.05	43.76	43.97	<b>45.35</b>	44.28
Dr GRPO	<b>45.48</b>	44.31	43.14	44.87	44.45
DISCO	43.98	<b>44.83</b>	<b>44.50</b>	45.07	<b>44.60</b>
<i>LLaMA3.2-1B</i>					
Base	22.31				
Naive GRPO	27.61	29.94	21.76	25.86	26.29
Dr GRPO	22.24	29.72	22.17	29.74	25.97
DISCO	<b>28.97</b>	<b>30.02</b>	<b>22.19</b>	<b>30.56</b>	<b>27.94</b>
<i>Qwen1.5-MoE</i>					
Base	45.57				
Naive GRPO	66.05	64.03	<b>66.04</b>	65.46	65.40
Dr GRPO	62.46	64.85	64.50	65.58	64.35
DISCO	<b>66.37</b>	<b>65.59</b>	65.19	<b>66.00</b>	<b>65.79</b>
<i>Qwen3-0.6B</i>					
Base	25.59				
Naive GRPO	41.99	34.83	34.72	34.31	36.46
Dr GRPO	42.08	33.23	36.76	32.91	36.25
DISCO	<b>45.07</b>	<b>35.64</b>	<b>44.44</b>	<b>38.28</b>	<b>40.86</b>

Table 4: Average accuracy under each domain-heavy training setup for different methods. Scores are averaged over five task-specific datasets per domain. Bold indicates the best-performing method.

the average (64.97 vs 64.91 for DISCO), though DISCO outperforms both baselines under the NQ-heavy and ARC-heavy conditions individually.

While Dr. GRPO occasionally surpasses Naive GRPO, the consistent advantages of DISCO, most notably the substantial gains on Qwen3-0.6B, LLaMA3.2-1B, and Qwen2.5-0.5B, underscore the effectiveness of incorporating explicit domain and difficulty signals. Statistical testing further confirms this improvement: compared to Naive GRPO, DISCO achieved a  $t$ -statistic of 2.97 with a one-tailed  $p$ -value of 0.0018, and compared to Dr. GRPO, it yielded a  $t$ -statistic of 3.49 with a one-tailed  $p$ -value of 0.0003 (Gould, 2010). Together, these results highlight the robustness of our joint scaling approach in enhancing GRPO alignment, particularly in mitigating performance trade-offs introduced by domain imbalance.

**Performance Breakdown by Dataset.** To illustrate how our method navigates domain trade-offs, we break down Qwen3-0.6B performance by evaluation dataset under each domain-heavy setting (Figure 2). This case study compares Naive GRPO,

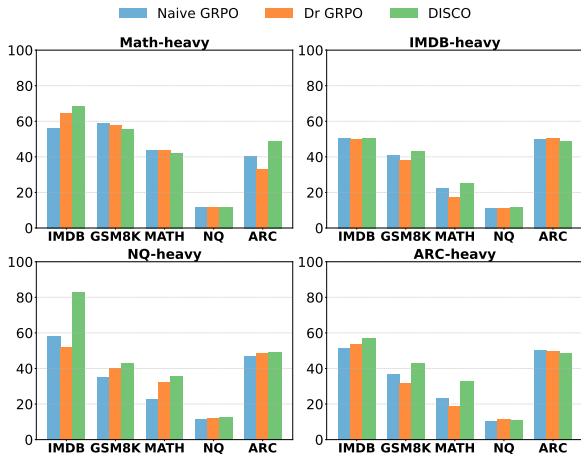


Figure 2: Qwen3-0.6B performance (EM Accuracy %) breakdown by the dataset under four domain-heavy training conditions (one condition per panel/subplot). Bar groups show results on individual datasets.

Dr. GRPO, and DISCO on the individual datasets.

The results for Qwen3-0.6B reveal a clear pattern where DISCO significantly boosts minority domain performance, sometimes involving a trade-off with majority domain scores. On majority domains (those comprising 75% of training data), DISCO performance varies compared to Naive GRPO. For instance, when trained NQ-heavy, DISCO improves performance on the NQ dataset (12.30% vs 11.47%). However, when trained Math-heavy, it scores lower on both MATH (41.68% vs 43.40%) and GSM8K (55.20% vs 58.91%). Similarly, under ARC-heavy training, the ARC score is slightly lower (48.29% vs 49.91%).

In contrast, DISCO demonstrates substantial and consistent improvements on tail domains. The most dramatic gains are seen in the NQ-heavy setting: the IMDB score jumps to 82.60% (from 57.90% for Naive GRPO), GSM8K increases to 42.72% (from 34.86%), and MATH rises to 35.70% (from 22.34%). Significant minority domain recovery is also evident in other settings, such as the gains on GSM8K (+2.28%) and MATH (+2.34%) under IMDB-heavy training.

These Qwen3-0.6B results show our joint scaling strategy effectively counters dominant domain overfitting, leading to significant recovery on minority domains, sometimes at the cost of peak head-domain performance. This yields a more balanced, generalized performance profile across diverse tasks, driven by the synergy between domain-aware reweighting and difficulty-aware scaling.

Model	Math-heavy	IMDB-heavy	NQ-heavy	ARC-heavy	Avg.
<i>Qwen2.5 0.5B</i>					
Naive GRPO	46.22	46.21	46.43	45.80	46.17
Diff only	46.87	46.21	45.03	46.07	46.05
Domain only	45.94	46.33	45.28	46.10	45.91
<b>DISCO</b>	<b>47.92</b>	<b>47.67</b>	<b>47.91</b>	<b>47.70</b>	<b>47.80</b>
<i>Qwen2.5 1.5B</i>					
Naive GRPO	64.81	64.91	65.15	65.02	64.97
Diff only	64.45	64.40	64.67	<b>65.39</b>	64.73
Domain only	<b>65.39</b>	64.73	64.65	65.18	<b>64.99</b>
<b>DISCO</b>	64.12	<b>64.93</b>	<b>65.26</b>	65.35	64.92
<i>LLaMA3.2 1B</i>					
Naive GRPO	27.61	29.94	21.76	25.86	26.29
Diff only	28.67	29.87	<b>22.46</b>	30.52	27.88
Domain only	26.42	29.61	21.87	30.01	26.98
<b>DISCO</b>	<b>28.97</b>	<b>30.02</b>	22.19	<b>30.56</b>	<b>27.94</b>

Table 5: Ablation study comparing Naive GRPO with variants using only difficulty-aware scaling (‘Diff only’), only domain-aware log-scaling (‘Domain only’, v1 weights), and both (‘DISCO’). Best result among the three variants in each numerical column is bolded.

### 5.3 Ablation Study of Scaling Components

#### Combining domain- and difficulty-aware scaling delivers the strongest overall performance.

To understand the individual contributions of our proposed scaling strategies, we conduct an ablation study selectively applying either the domain-aware weight (‘Domain only’) or the difficulty-aware weight (‘Diff only’), comparing against the full version (‘DISCO’) and the Naive GRPO baseline. Results are reported in Table 5.

We observe varied effects from the individual components across models. For instance, on Qwen2.5-0.5B, neither difficulty-aware scaling alone (46.05%) nor domain-aware scaling alone (45.91%) improved upon Naive GRPO (46.17%) on average. However, combining both in DISCO yields a significant boost to 47.80%. Conversely, on LLaMA3.2-1B, both ‘Diff only’ (27.88%) and ‘Domain only’ (26.98%) offer improvements over Naive GRPO (26.29%), and DISCO achieves the highest overall score (27.94%).

Interestingly, on the larger Qwen2.5-1.5B model, using ‘Domain only’ scaling achieves the highest average score (64.99%), slightly surpassing both Naive GRPO (64.97%) and DISCO (64.92%). This suggests domain re-weighting alone can be particularly effective for this model configuration. Despite ‘Domain only’ having the best average here, DISCO (64.92%) achieves competitive overall performance and secures the best scores among the variants under the IMDB-heavy and NQ-heavy conditions specifically.

These results reveal a complementary relationship between the two scaling components. Relying on one component alone proves insufficient: ‘Domain only’ scaling neglects sample difficulty variance within domains, while ‘Diff only’ scaling ignores global domain imbalance, both leading to inconsistent performance. In contrast, their joint use in DISCO consistently provides a more robust and generally higher-performing configuration across different models and domain imbalances, validating our combined approach.

#### 5.4 Effect of Group Size on Alignment Quality

To further test the robustness of DISCO, we vary the group size  $G$  used during GRPO and evaluate its impact. Figure 3 shows the performance of Naive GRPO and DISCO across  $G = \{2, 4, 8, 16\}$  on LLaMA3.2-1B under different training settings.

**Our method generalizes well across different group sizes, outperforming the baseline.** Across all setups, we observe that increasing group size generally improves performance for both methods. DISCO consistently outperforms Naive GRPO across almost all group sizes, highlighting its strong generalization ability. For example, under the Math-heavy setup, DISCO improves the average score from 20.06% to 23.11% at group size 2, and from 30.51% to 31.05% at group size 16. Similarly, in the ARC-heavy setup, DISCO reaches 31.88% compared to 30.64% for Naive GRPO at the largest group size. These results demonstrate that the benefits of reward scaling are robust to the group size  $G$ , making the approach broadly applicable within GRPO frameworks. Furthermore, this robustness extends to dataset size variations, as our method maintains consistent advantages when trained on 2,000 examples, detailed in Appendix B.

## 6 Related Work

**Reinforcement Learning from Human Feedback (RLHF)** has emerged as a powerful paradigm for aligning LLMs with human preferences and values. The most popular approach is Proximal Policy Optimization (PPO) (Ouyang et al., 2022; Schulman et al., 2017). Building upon PPO, GRPO eliminates the need for value functions through estimation from group scores and has emerged as a popular training paradigm (Shao et al., 2024). Rule-based GRPO further streamlines training by using predefined rules to evaluate outputs, avoiding the need for a separate reward model (Guo et al., 2025). Vari-

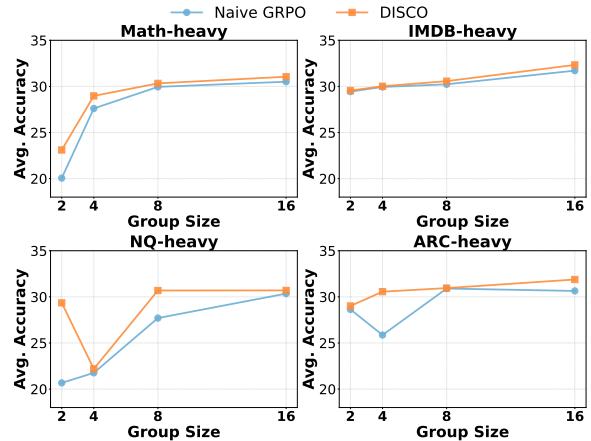


Figure 3: Impact of group size on LLaMA3.2-1B performance (EM Accuracy %) under different domain-heavy training distributions. Each line shows results for Naive GRPO and DISCO as group size varies.

ants address specific limitations of GRPO have also been proposed (Su et al., 2025). To mitigate length bias, Dr. GRPO removes normalization terms while SRPO introduces two-stage training and historical sampling (Liu et al., 2025; Zhang et al., 2025b). To address training instability, DAPO increases clip rates, and GVPO offers an analytical solution to KL-constrained reward maximization (Yu et al.; Zhang et al., 2025a). While prior work focuses on training challenges, we examine its vulnerability to dataset imbalance. Our proposed strategies are complementary and can be integrated with existing GRPO variants for improved performance.

**Data Imbalance in NLP** is a pervasive challenge characterized by significantly skewed distributions in training data. This issue is prevalent across various NLP tasks and often results in models that are biased toward the majority class, leading to poor performance on minority classes (Henning et al., 2023; He and Garcia, 2009; Liu et al., 2019; Wang et al., 2017; Godbole and Jia, 2022; Dai et al., 2023; Liu et al., 2024). Approaches to addressing data imbalance include data augmentation and loss adjustment methods (Zhou et al., 2024; Sun et al., 2007; Kandpal et al., 2023; Zhang et al., 2022; Tepper et al., 2020; Li et al., 2020; Kuo et al., 2024). Our work aligns with the loss adjustment paradigm. However, instead of reweighting loss, we adjust the rewards used in GRPO. While data augmentation techniques, such as generating synthetic data, can rebalance data distributions, they often introduce additional costs. In contrast, we focus on developing policy modifications to the GRPO process without incurring extra costs. Notably, DISCO can



be combined with data augmentation techniques.

## 7 Conclusion

In this work, we investigate the vulnerability of GRPO to domain imbalance in multi-domain alignment and introduce DISCO that combines domain and difficulty-aware reward scaling to enable GRPO to better handle skewed training distributions. Through experiments across diverse model scales, group sizes, and alignment settings, we demonstrate that DISCO consistently improves generalization and outperforms existing baselines. Ablation studies further confirm that both scaling components are essential. Overall, our method provides a robust enhancement to GRPO for aligning LLMs with real-world, imbalanced data.

## 8 Limitation

While our proposed domain- and difficulty-aware scaling demonstrates promising results for enhancing GRPO’s robustness to domain imbalance, this study has several limitations that suggest avenues for future research. Firstly, our experiments primarily utilized deterministic, rule-based rewards (Exact Match) to isolate the effects of the optimization dynamics. Real-world RLHF often involves complex, potentially noisy, and continuous rewards learned from human preferences. The efficacy and stability of our scaling methods under such learned reward functions warrant further investigation.

Secondly, we assumed the availability of clear, predefined domain labels for applying domain-aware scaling. The practical application to datasets with ambiguous, overlapping, or fine-grained domain structures, potentially requiring automatic domain identification, remains an open challenge. Furthermore, while we evaluated across diverse architectures and sizes up to the 14B scale (MoE), verification of performance consistency on significantly larger foundation models may still be needed.

Our methodological choices also present areas for future exploration. The reliance on self-consistency within a sampled group as a proxy for prompt difficulty is one of several possible heuristics. This work focused exclusively on modifying the GRPO algorithm’s optimization dynamics through reward scaling and did not explore alternative or potentially complementary strategies for handling domain imbalance that operate at the data level, such as data augmentation techniques (e.g.,

oversampling rare domains or generating synthetic data). Methods like these could offer different trade-offs and could potentially be combined with our algorithmic approach. Moreover, the interaction between our scaling mechanisms and other crucial RLHF hyperparameters, such as the KL divergence coefficient or learning rates, was not exhaustively studied.

Finally, our current evaluation was based on four predefined domain-heavy training settings, and performance was typically assessed using an unweighted average across different domain datasets. Future research could explore a more continuous spectrum of domain imbalance ratios by dynamically sampling data to construct a wider variety of training datasets. This would also allow for evaluation using different weighted averaging schemes for domain-specific scores, enabling a more nuanced understanding and potentially the mapping of Pareto-optimal frontiers in multi-domain performance.

## 9 Acknowledgment

Zhou, Liu, Wang and Huang are supported by DARPA Transfer from Imprecise and Abstract Models to Autonomous Technologies (TIAMAT) 80321, DARPA HR001124S0029-AIQ-FP-019, DOD-AFOSR-Air Force Office of Scientific Research under award number FA9550-23-1-0048, National Science Foundation NSF-IIS-2147276FAI, National Science Foundation NAIRR240045, National Science Foundation TRAILS Institute (2229885). Private support was provided by Peraton.

## References

- 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:1803.05457v1*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, and Yongbin Li. 2023. Long-tailed question answering in an open world. *arXiv preprint arXiv:2305.06557*.
- Ameya Godbole and Robin Jia. 2022. Benchmarking long-tail generalization with likelihood splits. *arXiv preprint arXiv:2210.06799*.

- Robert Gould. 2010. Statistics and the modern student. *International Statistical Review*, 78(2):297–315.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Haibo He and Eduardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–540.
- Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.
- Hsun-Yu Kuo, Yin-Hsiang Liao, Yu-Chieh Chao, Wei-Yun Ma, and Pu-Jen Cheng. 2024. Not all llm-generated data are equal: Rethinking data weighting in text classification. *arXiv preprint arXiv:2410.21526*.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.
- Ming Li, Lichang Chen, Jiu Hai Chen, Shwai He, Heng Huang, Jiuxiang Gu, and Tianyi Zhou. 2023. Reflection-tuning: Data recycling improves llm instruction-tuning. *arXiv preprint arXiv:2310.11716*.
- Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced nlp tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476.
- Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Hao-liang Wang, Tong Yu, et al. 2024. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.
- Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546.
- Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. [2 OLMo 2 Furious](#).
- OpenAI. 2023. [Gpt-4 technical report](#).
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- 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.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

- Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv e-prints*, pages arXiv–2503.
- Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern recognition*, 40(12):3358–3378.
- Gemma Team. 2024. [Gemma](#).
- Naama Tepper, Esther Goldbraich, Naama Zwerdling, George Kour, Ateret Anaby Tavor, and Boaz Carmeli. 2020. Balancing via generation for multi-class text classification improvement. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1440–1452.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, et al. 2025. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*.
- Chaoqi Wang, Zhuokai Zhao, Chen Zhu, Karthik Abinav Sankararaman, Michal Valko, Xuefei Cao, Zhaorun Chen, Madian Khabza, Yuxin Chen, Hao Ma, et al. 2024. Preference optimization with multi-sample comparisons. *arXiv preprint arXiv:2410.12138*.
- 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*.
- Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. *Advances in neural information processing systems*, 30.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhen-guo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL <https://arxiv.org/abs/2503.14476>.
- Chen Zhang, Lei Ren, Jingang Wang, Wei Wu, and Dawei Song. 2022. Making pretrained language models good long-tailed learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3298–3312.
- Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. 2025a. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*.
- Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, et al. 2025b. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.
- Yuhang Zhou, Jing Zhu, Paiheng Xu, Xiaoyu Liu, Xiyao Wang, Danai Koutra, Wei Ai, and Furong Huang. 2024. Multi-stage balanced distillation: Addressing long-tail challenges in sequence-level knowledge distillation. *Findings of EMNLP*.

## A Implementation Details

We train models using the OpenRLHF framework (Hu et al., 2024) with GRPO optimization. Each model is fine-tuned for 1 epoch. The rollout and training batch sizes are both set to 64, with micro-batch sizes of 8 and 4, respectively. We use a maximum prompt and generation length of 1024 tokens. The KL penalty is initialized at  $1e-3$  and estimated using the K3 estimator. All models use a learning rate of  $1e-6$ .

All evaluations are conducted using zero-shot inference, with models generating answers without access to in-context examples. For each task,

we apply a fixed, task-specific prompt template for both training and evaluation. The prompts are designed to clearly define the task instruction and input format.

Table 6 presents the prompt templates used across the five datasets.

Dataset	Prompt Template
MATH	Below is a math problem. Provide a detailed, step-by-step solution. ### Problem: {problem} ### Answer:
GSM8K	Same as MATH
IMDB	Below is a movie review. Determine the sentiment of the review as Positive or Negative. ### Review: {review} ### Answer:
NQ	Below is a question that requires a concise and accurate answer. Provide a detailed explanation before concluding with the correct answer. ### Question: {question} ### Answer:
ARC	Below is a question with multiple-choice answers. Choose the correct option based on your reasoning. ### Question: {question} ### Choices: A. {choice_A} B. {choice_B} ... ### Answer:

Table 6: Prompt templates used for zero-shot inference. Each template is applied consistently during both training and evaluation.

## B Effect of Reduced Dataset Size

To assess the robustness of our proposed scaling method to varying amounts of alignment data, we conducted additional experiments using a smaller dataset size of 2,000 examples (compared to 4,000 in the main experiments). We evaluated performance on the Qwen3-0.6B model under the four domain-heavy training distributions ( $G=4$ ).

The detailed results are presented in Table 7. Comparing the average performance (‘Avg.’ column) across the heavy training settings, we find that DISCO consistently outperforms both Naive GRPO and Dr. GRPO even with this reduced dataset. Averaged across the four heavy conditions, our method achieves a score of 33.97%, representing gains of +3.47% over Naive GRPO (30.50%) and +3.20% over Dr. GRPO (30.77%).

Notably, significant improvements are still observed on tail domains. For instance, under IMDB-heavy training, DISCO substantially improves GSM8K (+12.05% vs Naive) and MATH (+3.40% vs Naive) performance. While the absolute scores are generally lower than those achieved with 4,000 data (as expected), the relative advantage and the effectiveness of the domain- and difficulty-aware scaling mechanism persist. This supports the generalization ability of our method across different dataset sizes.

Training Dist.	Method	IMDB	GSM8K	MATH	NQ	ARC	Avg.
<i>Qwen3-0.6B (2K Training Data)</i>							
Math heavy	Naive GRPO	62.00	<b>52.72</b>	39.66	11.05	37.12	40.51
	Dr. GRPO	67.80	50.98	<b>41.18</b>	10.99	35.58	41.31
	DISCO	<b>75.60</b>	49.67	33.96	<b>11.33</b>	<b>47.01</b>	<b>43.51</b>
IMDB heavy	Naive GRPO	52.20	12.07	11.32	10.50	43.94	26.01
	Dr. GRPO	51.20	11.21	12.04	10.72	44.28	25.89
	DISCO	<b>55.50</b>	<b>24.12</b>	<b>14.72</b>	<b>11.05</b>	<b>46.58</b>	<b>30.39</b>
NQ heavy	Naive GRPO	57.90	11.03	10.50	<b>11.91</b>	44.11	26.79
	Dr. GRPO	56.40	12.82	11.00	11.55	44.62	27.28
	DISCO	<b>82.60</b>	<b>20.02</b>	<b>13.40</b>	10.94	<b>46.58</b>	<b>34.71</b>
ARC heavy	Naive GRPO	54.40	13.96	13.00	<b>11.14</b>	<b>51.02</b>	28.70
	Dr. GRPO	54.40	14.44	12.60	11.11	50.34	28.58
	DISCO	<b>55.20</b>	<b>32.66</b>	<b>19.42</b>	11.05	48.37	<b>33.34</b>

Table 7: Detailed results for Qwen3-0.6B trained with a reduced dataset size (2K examples,  $G=4$ ). Scores are EM accuracy (%) on individual datasets and the average, comparing alignment methods across domain-heavy training distributions.

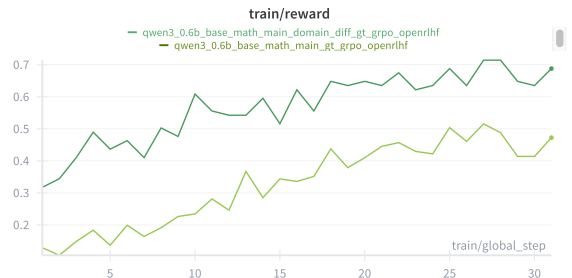


Figure 4: Training reward curves of Qwen3-0.6B models on math heavy training data for Naive GRPO (lighter green line) and DISCO (darker green line).

## C Supplementary Material

### C.1 Detailed Results on Individual Datasets

Detailed results for Table 4 in Section 5.2 are shown in Table 8.

### C.2 Reward Curve

To better understand how each component contributes during training, we visualize the reward trajectories of Naive GRPO and DISCO on Qwen3-0.6B in Figure 4.



Training Dist.	Method	IMDB	GSM8K	MATH	NQ	ARC	Avg.
<i>Qwen2.5 0.5B</i>							
Math heavy	Naive GRPO	87.80	58.98	24.38	13.96	45.99	46.22
	DISCO-Log	90.40	59.32	29.68	13.85	46.33	47.92
	Dr. GRPO	90.30	60.24	24.56	13.91	45.50	46.90
IMDB heavy	Naive GRPO	90.90	56.88	23.54	13.85	45.90	46.21
	DISCO-Log	89.60	57.88	29.86	14.43	46.59	47.67
	Dr. GRPO	91.30	56.96	23.54	13.99	45.30	46.22
NQ heavy	Naive GRPO	89.40	56.22	23.36	16.57	46.58	46.43
	DISCO-Log	91.30	57.29	29.92	15.29	45.73	47.91
	Dr. GRPO	90.40	56.65	29.26	16.89	45.98	47.84
ARC heavy	Naive GRPO	88.80	55.29	22.46	14.16	48.29	45.80
	DISCO-Log	90.60	56.48	29.44	14.04	47.95	47.70
	Dr. GRPO	90.60	56.12	29.54	13.32	48.54	47.62
<i>Qwen2.5 1.5B</i>							
Math heavy	Naive GRPO	93.80	84.60	52.34	25.98	67.32	64.81
	DISCO-Log	93.10	84.02	51.90	26.57	65.02	64.12
	Dr. GRPO	92.90	84.76	52.16	25.40	68.56	64.76
IMDB heavy	Naive GRPO	94.10	83.58	51.30	25.87	69.71	64.91
	DISCO-Log	93.20	83.54	50.86	25.10	71.93	64.93
	Dr. GRPO	93.20	83.78	51.10	24.56	71.10	64.75
NQ heavy	Naive GRPO	93.60	83.52	50.80	28.28	69.54	65.15
	DISCO-Log	92.90	83.50	50.32	27.76	71.84	65.26
	Dr. GRPO	92.60	83.41	50.26	27.56	69.79	64.72
ARC heavy	Naive GRPO	93.40	83.44	50.70	24.45	73.12	65.02
	DISCO-Log	93.10	83.22	50.40	26.57	73.46	65.35
	Dr. GRPO	93.30	83.45	50.49	25.42	73.12	65.16
<i>Qwen2-0.5B</i>							
Math heavy	Naive GRPO	90.80	76.53	6.28	12.66	33.96	44.05
	DISCO-Log	92.00	76.23	6.38	13.38	31.91	43.98
	Dr. GRPO	91.70	76.90	6.16	13.21	39.42	45.48
IMDB heavy	Naive GRPO	91.70	72.90	7.44	13.68	33.11	43.76
	DISCO-Log	90.40	74.66	7.64	13.46	37.97	44.83
	Dr. GRPO	90.50	74.29	7.00	12.69	37.05	44.31
NQ heavy	Naive GRPO	91.70	72.51	7.92	14.29	33.45	43.97
	DISCO-Log	91.50	74.70	7.20	13.49	35.60	44.50
	Dr. GRPO	91.50	72.54	8.02	13.85	29.78	43.14
ARC heavy	Naive GRPO	91.20	74.12	6.90	12.99	41.55	45.35
	DISCO-Log	90.70	74.27	6.68	13.16	40.53	45.07
	Dr. GRPO	89.70	72.63	8.18	12.35	41.51	44.87
<i>Qwen1.5-MoE-A2.7B</i>							
Math heavy	Naive GRPO	92.80	91.22	29.16	44.88	72.18	66.05
	DISCO-Log	93.70	92.05	29.46	43.68	72.95	66.37
	Dr. GRPO	94.50	92.06	29.42	45.65	50.68	62.46
IMDB heavy	Naive GRPO	95.40	83.50	24.80	43.57	72.86	64.03
	DISCO-Log	94.90	87.22	26.30	46.04	73.46	65.59
	Dr. GRPO	94.10	87.16	25.74	45.15	72.10	64.85
NQ heavy	Naive GRPO	94.50	90.03	26.84	46.48	72.35	66.04
	DISCO-Log	94.80	90.24	27.26	44.70	68.94	65.19
	Dr. GRPO	94.40	85.41	24.28	46.39	72.01	64.50
ARC heavy	Naive GRPO	94.70	86.61	27.20	43.88	74.91	65.46
	DISCO-Log	94.70	90.34	27.16	43.93	73.91	66.00
	Dr. GRPO	94.70	88.44	26.22	45.31	73.72	65.68
<i>Llama3.2 1B</i>							
Math heavy	Naive GRPO	75.60	3.25	3.94	21.72	33.53	27.61
	DISCO-Log	85.10	3.06	3.06	21.14	32.51	28.97
	Dr. GRPO	50.70	2.79	3.16	22.02	32.51	22.24
IMDB heavy	Naive GRPO	87.30	3.86	4.02	19.97	34.56	29.94
	DISCO-Log	88.80	3.14	3.34	20.97	33.87	30.02
	Dr. GRPO	88.00	3.54	3.50	20.47	33.07	29.72
NQ heavy	Naive GRPO	50.30	2.85	2.96	23.19	29.52	21.76
	DISCO-Log	51.10	2.54	3.08	22.66	31.57	22.19
	Dr. GRPO	50.90	2.87	2.88	22.29	31.93	22.17
ARC heavy	Naive GRPO	66.60	2.69	2.98	20.19	36.86	25.86
	DISCO-Log	87.90	3.31	3.74	19.02	38.83	30.56
	Dr. GRPO	86.10	3.40	3.34	19.78	36.09	29.74
<i>Qwen3-0.6B</i>							
Math heavy	Naive GRPO	55.70	58.91	43.40	11.51	40.44	41.99
	DISCO-Log	68.50	55.20	41.68	11.61	48.38	45.07
	Dr. GRPO	64.50	57.69	43.54	11.72	32.94	42.08
IMDB heavy	Naive GRPO	50.30	40.85	22.40	11.05	49.57	34.83
	DISCO-Log	50.10	43.13	24.74	11.36	48.89	35.64
	Dr. GRPO	50.00	37.97	17.06	11.05	50.09	33.23
NQ heavy	Naive GRPO	57.90	34.86	22.34	11.47	47.01	34.72
	DISCO-Log	82.60	42.72	35.70	12.30	48.89	44.44
	Dr. GRPO	51.90	39.74	32.08	11.63	48.55	36.76
ARC heavy	Naive GRPO	51.30	36.85	23.10	10.39	49.91	34.31
	DISCO-Log	56.70	42.69	32.76	10.94	48.29	38.28
	Dr. GRPO	53.70	31.50	18.70	11.08	49.57	32.91

Table 8: Detailed results showing EM accuracy (%) on individual datasets and the average, comparing alignment methods across models and domain-heavy training distributions (G=4).

Model	Math-heavy	IMDB-heavy	NQ-heavy	ARC-heavy	Avg.
<i>Qwen2.5 7B</i>					
Naive GRPO	60.18	61.38	61.94	<b>60.88</b>	61.07
DISCO	<b>60.52</b>	<b>61.74</b>	<b>63.16</b>	60.85	<b>61.57</b>
<i>Gemma2-2b-it</i>					
Naive GRPO	<b>56.85</b>	57.29	56.78	56.21	56.78
DISCO	56.68	<b>57.33</b>	<b>56.96</b>	<b>56.51</b>	<b>56.87</b>
<i>Olmo2-1B</i>					
Naive GRPO	34.78	34.57	33.38	33.97	34.18
DISCO	<b>36.61</b>	<b>34.74</b>	<b>34.65</b>	<b>34.74</b>	<b>35.19</b>

Table 9: Performance comparison of Naive GRPO vs. DISCO across domains.