DGPO: Beyond Pairwise Preferences with Directional Consistent Groupwise Optimization

Mengyi Deng[1], **Zhiwei Li**[1], **Xin Li**[1], **Tingyu Zhu**[1], **Yulan Yuan**[1], **Zhijiang Guo**[1,2,†], **Wei Wang**[1,2,†]
[1]Information Hub, The Hong Kong University of Science and Technology (Guangzhou), China
[2]The Hong Kong University of Science and Technology, Hong Kong SAR
{mdeng974, zli404, xli420, tzhu619, yyuan202}@connect.hkust-gz.edu.cn,
zhijiangguo@hkust-gz.edu.cn, weiwcs@ust.hk

Abstract

Although Large Language Models (LLMs) have made remarkable progress, current preference optimization methods still struggle to align directional consistency while preserving reasoning diversity. To address this limitation, we propose *Directional-Groupwise Preference Optimization* (DGPO), a lightweight framework that aggregates supervision signals at the group level and explicitly models direction-aware alignment through multi-candidate comparisons. DGPO organizes forward and reverse question-answer instances into structured sets and optimizes a margin-based likelihood objective that separates coherent reasoning paths from inconsistent alternatives. This groupwise formulation captures richer relative information than pairwise objectives and reinforces consistency across diverse reasoning pathways. Empirical results show that our constructed reverse data yields a 3.2% average improvement across five benchmarks, while DGPO further delivers consistent gains across multiple datasets and model families, achieving average accuracy improvements of up to 3.6%. Our code and data are available at https://github.com/Demi-deng2/DGPO.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities across diverse language and reasoning tasks (OpenAI, 2023; Touvron et al., 2023; Li et al., 2025). However, aligning these models to reason faithfully and follow intended problem semantics remains a key challenge. Most existing alignment efforts focus primarily on forward reasoning, where conclusions are derived step by step from given premises—for example, computing the arithmetic mean of all three-digit palindromes (see Figure 1). In contrast, *reverse reasoning*, such as inferring the total sum from the given mean and count of palindromes, has received far less attention. Human cognition is inherently bidirectional: in problem solving, planning, or theorem proving, people routinely combine forward deduction with backward inference (Al-Ajlan, 2015; Newell et al., 1972; Jara-Ettinger, 2019). This motivates the integration of reverse reasoning as a complementary element, essential for achieving more robust and generalizable alignment in LLMs.

Recent work explores reverse supervision as a complement to forward reasoning. MathGenie (Lu et al., 2024) employs back-translation from solutions to generate new problems, while Reverse Thinking (Chen et al., 2024) demonstrates that paired forward and backward exemplars enhance commonsense reasoning. Optimization-focused studies such as OptiBench (Yang et al., 2024) extend reverse supervision across linear and nonlinear settings, with ReSocratic applying back-translation of demonstrations. Deng et al. (2025) explicitly trains on inverted reasoning trajectories, enabling models to learn bidirectional reasoning patterns rather than relying solely on forward chains. Broader paradigms further highlight this trend: Reason-from-Future (Xu et al., 2025) alternates between forward and backward chains to improve math accuracy; backward reconstruction aids causal hypothesis testing (Ranaldi et al., 2025); and reverse exemplars strengthen reasoning (Deb et al., 2024; Wang et al., 2025).

Despite recent advances, existing reverse-supervision methods rarely specify how models should *internalize* directional signals. Reverse exemplars distilled from teacher models are used without explicit directional cues, limiting models' ability to capture richer alignment signals. This challenge is further compounded by the *Reversal Curse* (Berglund et al., 2023), which shows that transformers often fail to generalize under task inversion due to unstable entity binding, even when trained on large-scale inverted data. Architectural remedies such as the Joint-Embedding Predictive

[†] Co-corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 39386–39401

Architecture (Wang and Sun, 2025) mitigate this via additional memory components, but their reliance on strong inductive biases and structural modifications limits scalability. Moreover, most approaches emphasize a single correct solution while overlooking the diversity of valid reasoning paths, reducing models' ability to generalize across distinct reasoning trajectories. These observations call for alignment strategies that jointly model *directional consistency* and *reasoning diversity* within a unified framework.

We address these challenges with *Directional-Groupwise Preference Optimization (DGPO)*, a framework that builds on a high-quality dataset by generating both forward and reverse supervision, organizing them into structured groups, and optimizing a margin-based likelihood objective. DGPO regulates group preferences through directional consistency, while its groupwise formulation further encourages diversity across reasoning paths. This design effectively distinguishes inconsistent reasoning from coherent alternatives, thereby strengthening the overall alignment signal. Our main contributions are summarized as follows.

- **Data Augmentation**: Starting from 817 curated problems in the LIMO dataset (Ye et al., 2025), we construct groups that contain both forward and reverse instances, with each problem represented by three alternative solution paths in each direction. Distillation on the reverse solutions yields an average accuracy improvement of 3.2% over the base model.
- **Directional-Groupwise Optimization**: We introduce DGPO, which organizes forward and reverse supervision into structured groups and optimizes a margin-based likelihood objective. This groupwise formulation enforces directional consistency while naturally incorporating diverse reasoning paths.
- **Empirical Evaluation**: DGPO achieves consistent accuracy gains of 1%–3.6% across diverse benchmarks. Additional experiments analyze how scaling the number of reverse groups affects alignment performance.

2 Related Work

Data-Efficient Reverse Supervision. Research has increasingly examined reverse reasoning as a complement to forward supervision, highlighting its data efficiency and potential for enhancing model alignment and reasoning diversity (see Appendix 8.1 for further discussion). MathGenie (Lu et al., 2024) generates new problems via solution-to-question back-translation, and RevThink (Chen et al., 2024) distills paired forward–backward exemplars. Deng et al. (2025) trains on inverted reasoning trajectories, enabling models to learn bidirectional reasoning patterns. Verification-oriented methods such as FOBAR (Jiang et al., 2024) and RCoT (Xue et al., 2023) adopt reverse formulations for self-checking, while R^3 (Xi et al., 2024) introduces a reverse curriculum in reinforcement learning. OptiBench (Yang et al., 2024) and its ReSocratic synthesis strategy construct problems by reversing from solutions. Additional directions include causal hypothesis testing through backward reasoning (Ranaldi et al., 2025), reverse-style abductive inference (Deb et al., 2024), and reverse exemplars for few-shot prompting and geometry reasoning (Deng et al., 2024; Wang et al., 2025). Despite the demonstrated potential of reverse data, its contribution to LLM alignment remains insufficiently studied, particularly in enhancing diversity across reasoning pathways.

Direct Preference Optimization Variants. Direct Preference Optimization (DPO) (Rafailov et al., 2023) aligns language models through a closed-form preference objective, avoiding explicit reward-model fitting and unstable online reinforcement learning. Subsequent variants modify either the calibration strategy or the margin structure: KTO (Ethayarajh et al., 2024) reframes alignment with prospect-theoretic gains and losses, β-DPO (Wu et al., 2024) replaces a fixed inverse-temperature with a dynamic β, and SimPO (Meng et al., 2024) removes the reference model through a normalized reward surrogate. Despite these advances, existing DPO variants remain fundamentally pairwise and do not explicitly model whether multiple candidate solutions are directionally coherent with the same problem instance.

Group Relative Policy Optimization (Shao et al., 2024) enhances alignment by operating at the group level, where aggregating multiple outputs provides more diverse preference signals and improves generalization. Several extensions build on this idea: TreeRPO (Yang et al., 2025b) improves GRPO by replacing sparse trajectory-level rewards with tree-sampled, step-level dense rewards, enabling more fine-grained optimization of intermediate reasoning steps; Posterior-GRPO (Fan et al.,

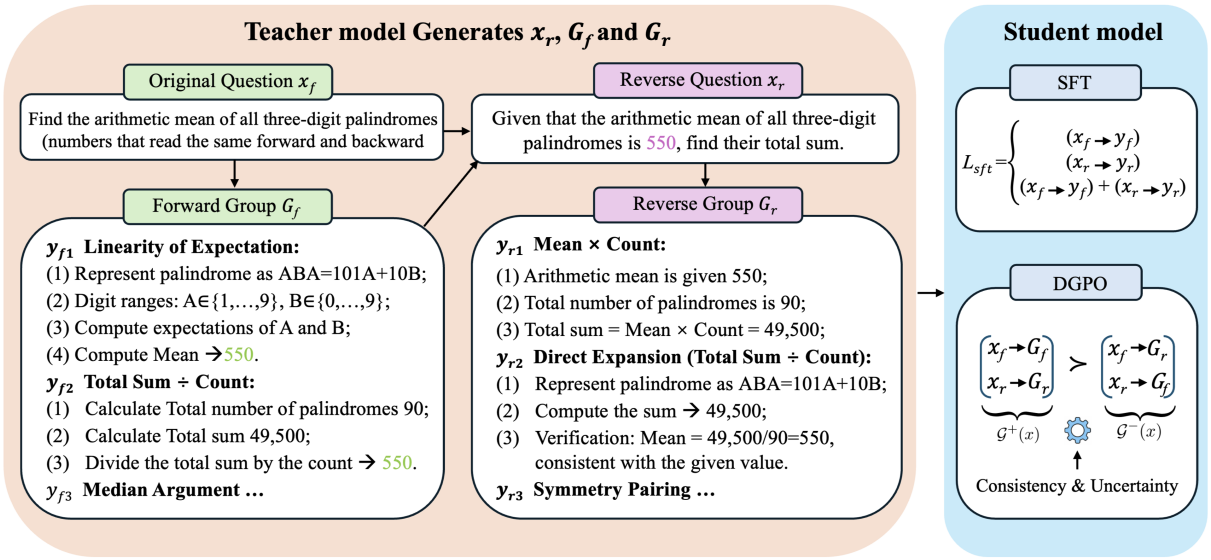

Figure 1: An overview of the DGPO training framework. The process begins with forward problems (x_f), each of which can be paired with a reverse question (x_r) formulated in the opposite reasoning direction. A teacher model then produces multiple candidate solutions for each problem type ($\{y_{fi}\}_{i=1}^3$ for x_f and $\{y_{ri}\}_{i=1}^3$ for x_r). The solutions are subsequently structured into direction-consistent ($\mathcal{G}^+$) and direction-divergent ($\mathcal{G}^-$) groups, wherein consistency is determined by matching a prompt's directionality with its corresponding solutions (e.g., x_f with $\{y_{fi}\}_{i=1}^3$). DGPO is trained on this structured supervision, incorporating directional modeling and uncertainty-based regulation to enhance alignment stability.

2025) conditions on successful outcomes to mitigate reward hacking and provide more reliable supervision; and DARS (Yang et al., 2025c) improves GRPO by correcting its bias via difficulty-adaptive rollout sampling and large-batch training to enhance both Pass@K and Pass@1 reasoning performance. Task-oriented refinements include GRPO-LEAD (Zhang and Zuo, 2025), which applies difficulty- and length-aware scaling to encourage concise mathematical reasoning, and noise-aware variants such as S-GRPO (Shen et al., 2025), which use advantage reweighting to improve robustness under imperfect supervision. Despite these advances, applications of group-based optimization to directional alignment remain scarce. This motivates our direction-aware extension, which constructs group-level supervision and leverages internal model judgments to enforce forward-reverse consistency while adaptively modulating update strength based on directional signals.

3 Methodology

We present *Directional-Groupwise Preference Optimization (DGPO)*, a training framework that aggregates supervision signals at the *group* level (see Figure 1). Starting from an original forward problem, the teacher model constructs its reverse counterpart and generates distinct solution paths for both directions, forming a coherent-direction group ($\mathcal{G}^+$) and a direction-inconsistent group ($\mathcal{G}^-$) (Section 3.1). The student model is then trained under DGPO, where Section 3.2 introduces directional consistency modeling, and Section 3.3 defines the groupwise training objective.

3.1 Directional Group Data Construction

We begin with the 817 curated problems, denoted as forward problems x_f, from the LIMO dataset (Ye et al., 2025), whose small scale and careful curation make it well-suited for studying reasoning alignment. For each x_f, the teacher model (DeepSeek V3 (Liu et al., 2024)) generates a corresponding reverse question x_r, as illustrated in Figure 1. Subsequently, Qwen3-32B (Yang et al., 2025a) solves both forward and reverse questions, producing three distinct forward solutions ($\{y_{fi}\}_{i=1}^3$) and three reverse solutions ($\{y_{ri}\}_{i=1}^3$), each with complete reasoning traces and final answers. To ensure reliability for downstream training, every solution is verified by Qwen3-8B: if an answer is judged incorrect, the corresponding reverse problem is resubmitted to Qwen3-32B until a correct solution is obtained.

Based on this pipeline, we organize the data into *directional groups* for DGPO training. For

each prompt x, we construct a set of preferred, direction-consistent solutions, $\mathcal{G}^+(x)$, and a set of dispreferred, direction-divergent solutions, $\mathcal{G}^-(x)$. Specifically, for a forward prompt x_f, the preferred set is composed of its corresponding forward solutions, $\mathcal{G}^+(x_f) = \{y_{fi}\}_{i=1}^3$, while the dispreferred set contains the solutions generated for its reverse counterpart, $\mathcal{G}^-(x_f) = \{y_{ri}\}_{i=1}^3$. Conversely, for a reverse prompt x_r, the preferred set consists of its validated reverse solutions, $\mathcal{G}^+(x_r) = \{y_{ri}\}_{i=1}^3$, and the dispreferred set is formed from the solutions of the original forward problem, $\mathcal{G}^-(x_r) = \{y_{fi}\}_{i=1}^3$. This bidirectional grouping strategy explicitly encodes the desired reasoning directionality, providing the structured preference signals required for DGPO training. The complete prompt templates and data collection configurations are detailed in Appendix 8.3.

3.2 Modeling Directional Consistency

Having constructed bidirectional groups of forward and reverse solutions, we train the model to discern directionally coherent reasoning from inconsistent alternatives. Each prompt includes multiple candidate responses, comprising direction-consistent solutions and reverse-problem solutions that appear off-target within the current reasoning context. Since these candidates reflect varying degrees of directional consistency, we introduce a confidence-aware mechanism that estimates directional alignment together with an associated uncertainty signal.

3.2.1 Consistency Estimation

For each prompt-solution pair (x, y), we process the concatenated sequence $(x; y)$ using the policy model π_θ to obtain a hidden representation $h_\theta(x, y)$. This representation is fed into a small projection head that predicts the parameters (α, β) of a Beta distribution. This distribution models the uncertainty in our estimate of the probability that solution y is directionally consistent with prompt x. We denote the resulting Beta-parameterized consistency distribution as:

$$q(d \mid x, y) = \mathrm{Beta}(\alpha(x, y), \beta(x, y)),$$

where $\alpha(x, y) > 0$ and $\beta(x, y) > 0$ are the positive outputs of the projection head. The mean of this distribution, $d(x, y) = \mathbb{E}[d] = \alpha/(\alpha + \beta)$, represents the estimated probability that the solution y is directionally consistent with the prompt x. The variance, $\sigma^2(x, y) = \mathrm{Var}[d] = \alpha\beta/[(\alpha + \beta)^2(\alpha + \beta + 1)]$, quantifies the uncertainty proxy in this estimate. To stabilize training, we prevent gradients from flowing from the consistency head back to the policy encoder's hidden states $h_\theta(x, y)$ using a stop-gradient operation.

3.2.2 Variance-Aware Aggregation

The estimated consistency and uncertainty are used to compute a weighted influence for each solution during group aggregation. While the framework theoretically supports multiple formulations for combining these signals, our implementation adopts a numerically stable variant where variance penalties are directly incorporated into the preference scores rather than the weights.

For each solution, we define the preference score as the length-normalized log-probability under the policy:

$$\Delta_\theta(y \mid x) \;=\; \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}), \quad (1)$$

where $|y|$ denotes the number of tokens in the response y. We compute the log-probability over response tokens only. This length normalization prevents systematic bias towards longer responses.

The pre-activation scores for the preferred and dispreferred groups are computed by combining the policy's likelihood, the estimated consistency, and an uncertainty penalty:

$$\begin{aligned} u^+(x, y) &= \tau_{\text{win}}^{-1} \Delta_\theta(y \mid x) + \log d(x, y) - \sigma^2(x, y), \\ u^-(x, y) &= \tau_{\text{lose}}^{-1} \Delta_\theta(y \mid x) + \log(1 - d(x, y)) - \sigma^2(x, y), \end{aligned} \quad (2)$$

where $\tau_{\text{win}}, \tau_{\text{lose}} > 0$ are temperature parameters. This formulation rewards solutions with high likelihood (Δ_θ), high predicted directional consistency ($\log d(x, y)$ or $\log(1 - d(x, y))$), and low estimated uncertainty (as directly penalized by the variance term $-\sigma^2(x, y)$).

3.3 DGPO Training Objective

The pre-activation scores for all solutions in a group are aggregated to form group-level scores using a temperature-scaled log-sum-exp operation:

$$A_\theta^+(x) = \tau_{\text{win}} \log \sum_{y \in \mathcal{G}^+(x)} \exp\left(u^+(x, y)\right), \quad (3)$$

$$A_\theta^-(x) = \tau_{\text{lose}} \log \sum_{y \in \mathcal{G}^-(x)} \exp\left(u^-(x, y)\right). \quad (4)$$

The core training objective is a contrastive loss that encourages a margin between the aggregated

score of the preferred group and that of the dispreferred group:

$$\mathcal{L}_{\text{DGPO}}(\theta, \phi) = -\mathbb{E}\left[\log \sigma\left(\lambda_{\text{margin}}\left(A_\theta^+(x) - A_\theta^-(x)\right)\right)\right], \tag{5}$$

where $\lambda_{\text{margin}} > 0$ is a scaling factor and σ is the logistic sigmoid function.

To mitigate model overconfidence and stabilize the training of the consistency head, we introduce two regularization terms. The first is a directional KL divergence penalty that incorporates prior knowledge of group structure by discouraging deviations from group-specific prior distributions:

$$\begin{aligned}\mathcal{R}_{\text{KL}} = \lambda_{\text{kl}} \mathbb{E}_{x,y}\big[&\mathbb{I}(y \in \mathcal{G}^+) \cdot \text{KL}\left(q \parallel p_+\right) \\ &+\mathbb{I}(y \in \mathcal{G}^-) \cdot \text{KL}\left(q \parallel p_-\right)\big],\end{aligned} \tag{6}$$

where $q = q(\cdot \mid x, y)$ denotes the learned consistency distribution, $\mathbb{I}(\cdot)$ is the indicator function, and p_+ and p_- are asymmetric Beta priors corresponding to the preferred and dispreferred groups, respectively. These priors are chosen such that p_+ is biased toward 1, encouraging high consistency estimates for $\mathcal{G}^+$, while p_- is biased toward 0, favoring low consistency estimates for $\mathcal{G}^-$.

The second is an optional penalty on the average predictive uncertainty across all candidate solutions for a prompt:

$$\mathcal{R}_{\text{var}} = \lambda_{\text{var}}^{(\text{grp})} \mathbb{E}_x \left[\frac{1}{|\mathcal{G}(x)|} \sum_{y \in \mathcal{G}(x)} \sigma^2(x, y)\right], \tag{7}$$

where $\mathcal{G}(x) \triangleq \mathcal{G}^+(x) \cup \mathcal{G}^-(x)$ denotes the full set of candidate solutions for prompt x.

The complete objective function minimized during training is:

$$\mathcal{J}(\theta, \phi) = \mathcal{L}_{\text{DGPO}}(\theta, \phi) + \mathcal{R}_{\text{KL}} + \mathcal{R}_{\text{var}}. \tag{8}$$

The parameters of the policy model θ and the consistency head ϕ are optimized jointly to minimize the objective $\mathcal{J}$. We employ the AdamW optimizer with standard practices, including gradient clipping and a cosine learning rate decay schedule. The base training procedure is summarized in Algorithm 1. The algorithm processes minibatches of prompts, and for each prompt, it computes the group aggregates and updates the model parameters. For specific implementation details, including hyperparameters and model configurations, we refer the reader to Section 8.2.

Algorithm 1 DGPO Training

Input: Minibatch $\mathcal{B}$, policy π_θ, consistency head q_ϕ, hyperparameters $\tau_{\text{win}}, \tau_{\text{lose}}, \lambda_{\text{kl}}, \lambda_{\text{var}}^{(\text{grp})}$
Output: Updated parameters θ, ϕ
for each prompt $x \in \mathcal{B}$ **do**
  Retrieve $\mathcal{G}^+(x)$ and $\mathcal{G}^-(x)$
  for each $y \in \mathcal{G}^+(x) \cup \mathcal{G}^-(x)$ **do**
    Compute $h_\theta(x, y)$
    Obtain $d(x, y)$, $\sigma^2(x, y)$ from q_ϕ
    Compute $\Delta_\theta(y \mid x)$ via Eq. (1)
    Form $u^+(x, y), u^-(x, y)$ via Eq. (2)
  end for
  Aggregate $A_\theta^+(x), A_\theta^-(x)$ via Eqs. (3)–(4)
  Compute $\mathcal{L}_{\text{DGPO}}(x)$ via Eq. (5)
  Compute $\mathcal{R}_{\text{KL}}(x)$ via Eq. (6)
  Compute $\mathcal{R}_{\text{var}}(x)$ via Eq. (7)
end for
$\mathcal{J} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} [\mathcal{L}_{\text{DGPO}}(x) + \mathcal{R}_{\text{KL}}(x) + \mathcal{R}_{\text{var}}(x)]$
Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{J}$
Update $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{J}$
return θ, ϕ

When each group $\mathcal{G}^+(x)$ and $\mathcal{G}^-(x)$ contains only one response, Eqs. (3)–(5) collapse to the familiar pairwise contrastive objective, showing that DGPO can be viewed as a groupwise extension of conventional preference optimization. When multiple responses are available, the log-sum-exp aggregation in Eqs. (3) and (4) acts as a smooth surrogate for selecting the higher-scoring responses from each group, allowing several candidates to contribute without introducing the discontinuity of a hard maximum. Finally, the Beta-parameterized consistency head provides a confidence-aware mechanism for modulating each response before group aggregation, so DGPO favors responses that are not only likely under the policy but also more aligned with the intended reasoning direction.

4 Experiments

4.1 Experimental Setup

Training Configuration. All experiments are implemented using the `SWIFT` framework (Zhao et al., 2024). We first perform supervised fine-tuning (SFT) for 3 epochs with a learning rate of 1×10^{-5} and a maximum sequence length of 11,000 tokens. Subsequently, DGPO training is conducted for 2

epochs with learning rates ranging from 1×10^{-6} to 3×10^{-6} and a maximum sequence length of 1,000 tokens. All models are trained in **bfloat16** precision on four NVIDIA A6000 GPUs (48GB each). Additional training details are provided in Appendix 8.5.

Evaluation. We employ five distinct benchmarks to thoroughly assess the reasoning alignment performance, covering both in-distribution performance and generalization capabilities. The OpenAI Math 500 dataset (Lightman et al., 2023; Hendrycks et al., 2021a) is adopted as an *in-domain* benchmark, as it closely aligns with the algebraic, geometric, and symbolic reasoning characteristics present in LIMO. AIME-25 (Committee, 2025) is considered a *near-domain* benchmark, sharing a competition-style format with LIMO while exhibiting differences in problem distribution and year-specific variations. To further evaluate generalization, we incorporate three *out-of-domain* benchmarks: GPQA (Rein et al., 2024), which assesses graduate-level scientific and conceptual reasoning; Gaokao MathQA (GMQ) (Zhang et al., 2023), comprising high-school level mathematical word problems; and Leaderboard Math: Math Geometry Hard (LMGH) (Hendrycks et al., 2021b), a curated set of challenging geometry problems. Collectively, these benchmarks encompass diverse domains, difficulty levels, and problem types, providing a comprehensive evaluation framework for reasoning alignment.

Our evaluation reports the pass@1 accuracy in a zero-shot chain-of-thought setting. We generate reasoning paths using greedy sampling, setting the maximum output length to 16,000 tokens for AIME-25 and 10,000 tokens for the other benchmarks to prevent truncation. The evaluation process follows the standard lm-eval-harness (Gao et al., 2024), leveraging its rule-based matching system that incorporates canonicalized string normalization and numerical equivalence verification. This consistent protocol guarantees an objective and fully reproducible evaluation across the diverse set of reasoning benchmarks.

4.2 Effectiveness of Group Data and DGPO

Before evaluating the effectiveness of DGPO, we first examine the quality and behavioral impact of the constructed reverse data through supervised fine-tuning (SFT). Experiments are conducted mainly on Qwen3-1.7B-Base. We distill models on the forward (LIMO) and reverse datasets independently, as well as on their mixed combinations, except for the 3Mixed setting, each containing problems with exactly one verified reasoning path. As shown in Table 1 and Appendix 8.6, the reverse subset data demonstrates comparable quality to the original forward LIMO data. For the Qwen3-1.7B-Base, SFT with reverse data achieves an average accuracy of 25.2%, slightly higher than forward-only SFT (24.2%), confirming that the reverse counterparts provide meaningful and learnable supervision.

However, directly mixing forward and reverse data results in weaker performance compared to single-direction SFT. The Mixed setting drops to 24.1%, and increasing the ratio of reverse data in the 3Mixed setting further reduces performance to 23.1%. This suggests that naive data blending introduces interference between opposite reasoning directions rather than synergy. A similar pattern appears for the Qwen3-1.7B model (see Appendix 8.6), where distillation on reverse subsets or mixed data causes noticeable drops across benchmarks. While the constructed groupwise data derived from forward and reverse subsets is of high quality, these results reveal that naive joint training across opposite directions introduces conflicts. This shows the need for an alignment methodology that can effectively leverage groupwise supervision while preserving the diversity of directional reasoning.

We evaluate DGPO on three model foundations: Qwen3-1.7B-Base, SFT model trained on the 3Mixed subsets, and Qwen3-1.7B combining forward and reverse data. DGPO is trained on 1,634 groupwise preference instances, where each forward and reverse counterpart form directional contrasted groups, enforcing consistency between preferred and dispreferred reasoning directions. As shown in Table 1, DGPO delivers consistent performance improvements across all model families. On Qwen3-1.7B-Base, DGPO improves the average accuracy from 22.0% to 24.7%, confirming that groupwise optimization provides alignment benefits beyond direct fine-tuning. When applied to the mixed-direction SFT model (3Mixed), DGPO further strengthens results by 2.2% and clearly outperforms vanilla DPO, demonstrating its effectiveness in mitigating inconsistencies introduced by multi-directional training.

For Qwen3-1.7B reported in Table 4, DGPO improves the average accuracy from 27.5% to 30.9%,

Table 1: Performance comparison of DGPO across two model families: Qwen3-1.7B-Base and SFT models trained on mixed forward–reverse data followed by DGPO. We also compare different SFT dataset configurations, where 1Mixed denotes equal-size blending of forward (LIMO) and reverse data, and 3Mixed triples the reverse portion. None denotes the official pretrained model without any additional fine-tuning or preference optimization. The best result is highlighted in **bold**, and the second best is underlined.

Base Model	Training Strategy	AIME-25	GPQA	Math 500	GMQ	LMGH	Avg. Acc.
Qwen3-1.7B-Base	None	0%	28.3%	45.8%	33.6%	2.3%	22%
Qwen3-1.7B-Base	SFT (LIMO)	0%	27.8%	47.2%	34.8%	11.4%	24.2%
Qwen3-1.7B-Base	SFT (Reverse)	3.3%	29.8%	46.2%	33.6%	**12.9%**	25.2%
Qwen3-1.7B-Base	SFT (1Mixed)	0%	29.3%	46.2%	**35.3%**	9.9%	24.1%
Qwen3-1.7B-Base	SFT (3Mixed)	0%	27.3%	43.2%	33.9%	11.3%	23.1%
SFT (3Mixed)	Vanilla DPO	0%	27.8%	39.2%	30.3%	4.5%	20.4%
SFT (3Mixed)	DGPO (Ours)	6.7%	**30.3%**	43.6%	33.6%	12.1%	**25.3%**
Qwen3-1.7B-Base	DGPO (Ours)	**10%**	28.8%	**46.6%**	35%	3.0%	24.7%

Table 2: Performance comparison of DGPO and other DPO variants on Qwen3-1.7B-Base. The best result in each metric is highlighted in **bold**.

Base Model	DPO Variant	AIME-25	GPQA	Math 500	GMQ	LMGH	Avg. Acc.
Qwen3-1.7B-Base	β-DPO	0%	27.3%	47.6%	35.6%	3.8%	22.9%
Qwen3-1.7B-Base	γ-DPO	0%	28.3%	45.6%	35.0%	1.5%	22.1%
Qwen3-1.7B-Base	SimPO	0%	27.8%	47.0%	33.3%	3.0%	22.2%
Qwen3-1.7B-Base	Ours	**10.0%**	**28.8%**	46.6%	35.0%	3.0%	**24.7%**

with substantial gains on AIME-25 and GPQA, indicating stronger generalization to near and out-of-distribution tasks. However, a slight decline on Math 500 suggests a modest trade-off for in-distribution performance as the model learns to balance broader reasoning consistency. These results demonstrate that DGPO effectively integrates groupwise supervision across reasoning directions and strengthens alignment even on highly optimized models. To further validate the robustness and reliability of our results, we repeat each DGPO experiment three times with different random seeds and report the mean accuracy and standard deviation across runs in Table 6.

As shown in Table 2, DGPO achieves the strongest average accuracy on Qwen3-1.7B-Base. Compared with β-DPO (Wu et al., 2024) and γ-DPO (Sun et al., 2025), DGPO consistently improves AIME-25 while remaining competitive on GPQA and GMQ, indicating that modeling directional consistency at the group level yields a more reliable alignment signal than dynamically rescaling pairwise margins alone. Compared with SimPO (Meng et al., 2024), DGPO also improves out-of-domain generalization on GPQA and maintains a stronger balance across benchmarks, suggesting that removing the reference model is insufficient without explicitly separating direction-consistent from direction-divergent reasoning paths.

Table 3 presents a detailed ablation analysis of the core components in DGPO. The base configuration (Ours) integrates both the variance regularization term $\mathcal{R}_{\text{var}}$ and a differentiable posterior head that explicitly models directional consistency within each group. When the posterior head is removed, the training reduces to an offline GRPO-style objective, where preference scores within each group are computed solely from the averaged log probabilities $\log \pi_\theta(y|x)$ and combined through a smooth maximum operation, without explicitly modeling directional alignment or uncertainty. Even though this simplified variant still brings moderate improvements, the overall gains are clearly smaller than those achieved by the full DGPO framework, with the most evident declines observed on AIME-25.

In contrast, removing the variance regularization term $\mathcal{R}_{\text{var}}$ leads to a moderate performance decrease of 1.8%, while removing the posterior head results in reductions of 2.9%. These findings suggest that penalizing predictive uncertainty helps

Table 3: Ablation study on different training settings. **Ours** employs variance regularization $\mathcal{R}_{\text{var}}$ and a differentiable posterior head. We ablate these components by removing the variance regularization (w/o $\mathcal{R}_{\text{var}}$) and by disabling the differentiable posterior (w/o Posterior). Percentages with ↓ indicate the average performance drop relative to **Ours**. None denotes the official model without further fine-tuning or alignment training. The best results within each base model family are highlighted in **bold**.

Base Model	Setting	AIME-25	GPQA	Math 500	GMQ	LMGH	Avg. Acc.
Qwen3-1.7B-Base	None	0%	28.3%	45.8%	33.6%	2.3%	22%
	Ours	**10%**	**28.8%**	46.6%	35%	3.0%	**24.7%**
	w/o $\mathcal{R}_{\text{var}}$	3.3%	**28.8%**	**46.8%**	**32.5%**	3.0%	22.9%(1.8%↓)
	w/o Posterior	0%	27.8%	45%	33.0%	3.0%	21.8%(2.9%↓)

Table 4: Effect of the number of reverse problem groups on model performance. **Number** indicates the group configuration: 0 denotes the pretrained model without additional training, 2 corresponds to the LIMO forward set paired with one reverse group, 3 with two, and 4 with three. Within each base model family, the best result is shown in **bold** and the second-best is underlined. An upward arrow (↑) indicates the relative improvement in average accuracy over the base model.

Base Model	Number	AIME-25	GPQA	Math 500	GMQ	LMGH	Avg. Acc.
Qwen3-1.7B-Base	0	0%	28.3%	45.8%	33.6%	2.3%	22%
	2	**10%**	28.8%	46.6%	35%	3.0%	**24.7%**(2.7%↑)
	3	3.3%	**29.8%**	47.0%	33.9%	3.8%	23.6%(1.6%↑)
	4	6.7%	29.3%	**47.8%**	**35.6%**	**4.5%**	23.9%(1.9%↑)
Qwen3-1.7B	0	13.3%	29.8%	47.6%	36.2%	10.6%	27.5%
	2	**26.7%**	32.3%	46%	36.5%	12.9%	30.9% (3.4%↑)
	3	20.0%	**34.8%**	**48.0%**	36.2%	**16.7%**	**31.1%**(3.6%↑)
	4	23.3%	32.8%	46.6%	**37%**	13.6%	30.7%(3.2%↑)

stabilize groupwise optimization and that directional consistency modeling provides informative signals for distinguishing coherent from divergent reasoning patterns. Both uncertainty regularization and explicit directional modeling are key components contributing to DGPO's stable and robust improvements over standard groupwise contrastive training.

4.3 Scaling Effects of Reverse Group Augmentation

To examine how the quantity of reverse problem groups influences alignment, we systematically vary the number of reverse sets paired with each forward exemplar. Each forward problem in the LIMO dataset can be paired with multiple reverse questions generated by DeepSeek V3, where each problem (forward or reverse) is accompanied by three native solutions produced by Qwen3-32B (see Appendix 8.3 for data construction details). For DGPO training, the preferred group $\mathcal{G}^+(x)$ for a given problem x consists of its direction-consistent solutions, while for the scaling experiment, the dispreferred group $\mathcal{G}^-(x)$ is formed by selecting direction-divergent solutions drawn from unrelated or mismatched problem instances. This design enables a controlled investigation of group-level augmentation while maintaining consistent supervision quality.

As shown in Table 4, the effect of increasing reverse groups exhibits distinct trends across model families. For Qwen3-1.7B-Base, moderate augmentation enhances alignment, but adding more groups leads to progressively smaller gains. Performance peaks when introducing a single reverse group, suggesting that mild directional diversification most effectively strengthens the model's reasoning alignment. However, further augmentation leads to a slight decline, implying that excessive directional mixing introduces representational interference and hinders stable generalization. This consistent decline in performance indicates that low-capacity base models struggle to accommodate increasingly diverse directional signals.

In contrast, the Qwen3-1.7B model exhibits a more stable and scalable improvement trend. Performance increases steadily with additional reverse

groups, reaching a peak of 31.1% when two reverse groups are included. Most of the gains come from GPQA, Math 500, and LMGH, implying that expanding reverse-group data particularly strengthens reasoning generalization. When the number of reverse groups increases further, only modest gains are observed on GMQ and LMGH, while the overall performance falls slightly below that of the two-group configuration, indicating that the benefits of additional augmentation are limited. Taken together, these results indicate that while expanding the number of reverse groups yields measurable improvements in reasoning alignment, the gains gradually saturate rather than continue to scale with data size.

5 Conclusion

This paper introduces DGPO, a framework that models directional consistency while preserving intra-group diversity. Built on curated forward–reverse exemplars, it constructs structured groups capturing coherent and contrastive reasoning behaviors for finer alignment. Empirical results demonstrate that DGPO consistently improves alignment across multiple model families, achieving accuracy gains of 2.2%–3.6%. Ablation studies confirm that both uncertainty regularization and direction-aware modeling are beneficial, as removing either leads to a 1.8%–3.6% reduction in average accuracy. Scaling analyses further reveal that moderate reverse-group augmentation enhances alignment robustness, whereas excessive augmentation yields diminishing returns due to over-diversification. Overall, DGPO offers a data-efficient approach that balances directional consistency and reasoning diversity across domains.

6 Limitations

While the proposed DGPO framework benefits from the bidirectional data construction that explicitly introduces forward–reverse reasoning pairs, our dataset generation pipeline is not entirely free from imperfection. Although the solutions are verified for correctness by a model, the constructed problems themselves are not strictly validated for logical or semantic reversibility. Consequently, a portion of the reverse problems may not represent true inverses of their forward counterparts, and some may even lack well-defined answers under the intended reasoning direction. Despite this limitation, DGPO remains robust in practice: its group-wise contrastive formulation and uncertainty-aware modeling effectively mitigate the influence of such imperfect data, leading to stable optimization and consistent performance gains.

7 Acknowledgement

This work is supported by the Advanced Materials-National Science and Technology Major Project (Grant No. 2025ZD0620100), HKUST(GZ)-IEIP-RoP (G01RF000256), and National Key R&D Program of China (No. 2024YFA1012700).

References

Ajlan Al-Ajlan. 2015. The comparison between forward and backward chaining. *International Journal of Machine Learning and Computing*, 5(2):106–113.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

M. Besta and 1 others. 2024. Graph-of-thoughts: Solving complex problems with llms. *arXiv preprint arXiv:2402.00000*.

Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, and 1 others. 2024. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

AIME Committee. 2025. Aime-25 dataset. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions. American Invitational Mathematics Examination 2025 problems.

Aniruddha Deb, Neeva Oza, Sarthak Singla, Dinesh Khandelwal, Dinesh Garg, and Parag Singla. 2024. Fill in the blank: Exploring and enhancing llm capabilities for backward reasoning in math word problems. In *Annual Meeting of the Association for Computational Linguistics*.

Linger Deng, Yuliang Liu, Bohan Li, Dongliang Luo, Liang Wu, Chengquan Zhang, Pengyuan Lyu, Ziyang Zhang, Gang Zhang, Errui Ding, and 1 others. 2024. R-cot: Reverse chain-of-thought problem generation for geometric reasoning in large multimodal models. *arXiv e-prints*, pages arXiv–2410.

Mengyi Deng, Xin Li, Tingyu Zhu, Zhicheng Yang, Zhijiang Guo, and Wei Wang. 2025. When inverse data outperforms: Exploring the pitfalls of mixed data in multi-stage fine-tuning. *arXiv preprint arXiv:2509.13079*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Lishui Fan, Yu Zhang, Mouxiang Chen, and Zhongxin Liu. 2025. Posterior-grpo: Rewarding reasoning processes in code generation. *arXiv preprint arXiv:2508.05170*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Dawn Tang, Rohan Desai, Dawn Song, and Jacob Steinhardt. 2021a. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Julian Jara-Ettinger. 2019. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110.

Weisen Jiang, Han Shi, and James T. Kwok. 2024. Forward-backward reasoning in large language models for mathematical verification. In *Findings of the Association for Computational Linguistics (ACL)*, pages 6647–6661.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Xiao Liang, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu Chen. 2025. Beyond pass@ 1: Self-play with variational problem synthesis sustains rlvr. *arXiv preprint arXiv:2508.14029*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Zimu Lu, Tong Wang, Hao Peng, Yitong Sun, Dong Yu, William Yang Wang, and Zhiting Hu. 2024. Mathgenie: Generating and verifying reasoning paths for math word problems. *arXiv preprint arXiv:2402.16352*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Allen Newell, Herbert Alexander Simon, and 1 others. 1972. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ.

OpenAI. 2023. Gpt-4 technical report. OpenAI Technical Report. https://openai.com/research/gpt-4.

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.

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.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Leonardo Ranaldi, Fabio Addino, and Andrea Bacciu. 2025. Exploring backward reasoning in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*.

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*.

T. Schick and 1 others. 2023. Toolformer: Language models can teach themselves to use tools. In *ICLR*.

Zhihong Shao, Zihan Liu, Wei Zhang, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Si Shen, Peijun Shen, Wenhua Zhao, and Danhao Zhu. 2025. Mitigating think-answer mismatch in llm reasoning through noise-aware advantage reweighting. *arXiv preprint arXiv:2508.05928*.

Jie Sun, Junkang Wu, Jiancan Wu, Zhibo Zhu, Xingyu Lu, Jun Zhou, Lintao Ma, and Xiang Wang. 2025. Robust preference optimization via dynamic target margins. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5399–5416.

Fengwei Teng, Quan Shi, Zhaoyang Yu, Jiayi Zhang, Yuyu Luo, Chenglin Wu, and Zhijiang Guo. 2025. Atom of thoughts for markov llm test-time scaling. *arXiv preprint arXiv:2502.12018*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Boshi Wang and Huan Sun. 2025. Is the reversal curse a binding problem? uncovering limitations of transformers from a basic generalization failure. *arXiv preprint arXiv:2504.01928*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Zhen Wang, Xi Zhou, Yating Yang, Bo Ma, Lei Wang, and Rui Dong. 2025. Sgeu: enhancing llm reasoning via backward exemplar generation and verification: Z. wang et al. *Applied Intelligence*, 55(10):748.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024. β-DPO: Direct Preference Optimization with Dynamic β. *Advances in Neural Information Processing Systems*, 37:129944–129966.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, and 1 others. 2024. Training large language models for reasoning through reverse curriculum reinforcement learning. *arXiv preprint arXiv:2402.05808*.

Yinlong Xu, Yanzhao Zheng, Shuoshuo Sun, Shuaihan Huang, Baohua Dong, Hangcheng Zhu, Ruohui Huang, Gang Yu, Hongxia Xu, and Jian Wu. 2025. Reason from future: Reverse thought chain enhances llm reasoning. In *Findings of the Association for Computational Linguistics (ACL)*.

Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. 2023. Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought. *arXiv preprint arXiv:2305.11499*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. 2025b. Treerpo: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Yongxin Wang, Dongchun Xie, Hanhui Li, Yiwei Wang, Xiaodan Liang, and Jing Tang. 2025c. Depth-breadth synergy in rlvr: Unlocking llm reasoning gains with adaptive exploration. *arXiv preprint arXiv:2508.13755*.

Zhicheng Yang, Yiwei Wang, Yinya Huang, Zhijiang Guo, Wei Shi, Xiongwei Han, Liang Feng, Linqi Song, Xiaodan Liang, and Jing Tang. 2024. Optibench meets resocratic: Measure and improve llms for optimization modeling. *arXiv preprint arXiv:2407.09887*.

Shunyu Yao, Nathanael Schärli, Nathan Scales, Jiahai Zhao, Jiameng Chen, Lei Hou, Denny Zhou, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Run-Ze Yang, Nan Cui, and Karthik Narasimhan. 2023b. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*.

Yue Zhang and Tianyu Zuo. 2025. GRPO-LEAD: A difficulty-aware rl approach for concise math reasoning. *arXiv preprint arXiv:2504.09696*.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2024. Swift:a scalable lightweight infrastructure for fine-tuning. *Preprint*, arXiv:2408.05517.

8 Appendix

8.1 Additional Related Work

Data Quality Over Scale in Reasoning Alignment. A recurring finding in reasoning alignment is that supervision quality and structural diversity often matter more than sheer scale. Less is More for Reasoning (LIMO) (Ye et al., 2025) demonstrates that as few as 817 carefully curated exemplars can yield substantial gains in reasoning performance and generalization. Comparable observations arise in efficiency-oriented scaling (Muennighoff et al., 2025), where smaller models achieve performance on par with much larger ones when supported by high-quality exemplars. At the inference level, diversification techniques such as Self-Consistency (Wang et al., 2022), Tree-of-Thoughts (Yao et al., 2023a), Graph-of-Thoughts (Besta et al., 2024), and Atom-of-Thoughts (Teng et al., 2025), together with tool-augmented approaches like ReAct (Yao et al., 2023b), Toolformer (Schick et al., 2023), and PAL (Gao et al., 2023), further demonstrate that aggregating diverse reasoning trajectories enhances robustness. On the training side, methods such as self-distillation (Zelikman et al., 2022), verifier-based filtering (Cobbe et al., 2021), Self-Play training instances synthesis (Liang et al., 2025) and feedback-driven frameworks including Constitutional AI (Bai et al., 2022; Ouyang et al., 2022) show that structured supervision can substitute for brute-force scaling. Nevertheless, most existing approaches remain confined to *forward-only* supervision, rely heavily on powerful teacher models, or incur substantial costs for filtering high-quality data.

8.2 Implementation in Experiments

Hidden Representation. For each concatenated sequence $(x; y)$, the hidden state corresponding to the last non-padding token in the final transformer layer is extracted as $h_\theta(x, y)$, identified via the attention mask. A stop-gradient operation is applied to the posterior features in all experiments to prevent gradient flow in subsequent computations.

Posterior Head. A linear projection layer maps the hidden state to two parameters (α, β) for each response. A softplus activation ensures the positivity of these parameters, forming a Beta posterior distribution $q(d \mid x, y) = \mathrm{Beta}(\alpha, \beta)$. The mean $d(x, y) = \alpha/(\alpha + \beta)$ and variance $\sigma^2(x, y) = \alpha\beta/[(\alpha + \beta)^2(\alpha + \beta + 1)]$ are derived from this distribution.

Preference Score. The preference score is defined as the length-normalized log-probability of the response tokens under the policy model, expressed as $\Delta_\theta(y \mid x) = \frac{1}{|y|} \log \pi_\theta(y \mid x)$. Notably, no reference model is employed in this setup, and thus no log-ratio term is included.

Variance-Informed Score Aggregation. Uncertainty is directly incorporated into the scoring mechanism through the pre-activation terms:

$$u^+ = \tau_{\text{win}}^{-1} \Delta_\theta + \log d - c\sigma^2,$$

$$u^- = \tau_{\text{lose}}^{-1} \Delta_\theta + \log(1 - d) - c\sigma^2,$$

with $\tau_{\text{win}} = 0.8$, $\tau_{\text{lose}} = 1.2$, and $c = 1.0$. These per-response scores are aggregated at the group level using a temperature-scaled log-sum-exp operation:

$$A_\theta^+ = \tau_{\text{win}} \log \sum_{y \in \mathcal{G}^+} e^{u^+},$$

$$A_\theta^- = \tau_{\text{lose}} \log \sum_{y \in \mathcal{G}^-} e^{u^-}.$$

Group Gating. In the experimental implementation, a gated margin is applied to modulate group-level scores:

$$g_{\text{win}}(x) = \frac{1}{|\mathcal{G}^+|} \sum_{y \in \mathcal{G}^+} d(x, y),$$

$$g_{\text{lose}}(x) = \frac{1}{|\mathcal{G}^-|} \sum_{y \in \mathcal{G}^-} (1 - d(x, y)).$$

A gated margin is then constructed as

$$M_\theta(x) = (A_\theta^+ - A_\theta^-) + \gamma \big[\log g_{\text{win}} - \log g_{\text{lose}}\big],$$

with $\gamma = 1.0$. The contrastive loss is formulated as $-\log \sigma(\lambda_{\text{margin}}\, M_\theta(x))$, where $\lambda_{\text{margin}} > 0$ is the logistic scaling factor.

KL Regularization. A directional Beta prior is applied to regularize the posterior: $\mathrm{Beta}(2, 1)$ for winning responses ($y \in \mathcal{G}^+$), encouraging $d \to 1$, and $\mathrm{Beta}(1, 2)$ for losing responses ($y \in \mathcal{G}^-$), encouraging $d \to 0$. The strength of the KL divergence penalty is controlled by a coefficient λ_{kl}.

Variance Penalty. An additional penalty term, weighted by a coefficient $\lambda_{\text{var}}^{(\text{grp})}$, is applied to the average posterior variance σ^2. When variance is already integrated into the score aggregation, this standalone penalty is reduced by an order of magnitude to prevent excessive regularization.

Beta Prior Parameters for Directional KL Penalty

The directional KL divergence penalty $\mathcal{R}_{\mathrm{KL}}$ defined in Equation 6 utilizes asymmetric Beta distributions as informative priors. The specific parameter choices are as follows:

- **Preferred group prior (p_+):** We use $\mathrm{Beta}(2, 1)$ for the preferred group $\mathcal{G}^+$. This distribution has a mode at $(2-1)/(2+1-2) = 1$ and mean $2/3 \approx 0.67$, placing strong probability mass near 1 to encourage high consistency estimates.
- **Dispreferred group prior (p_-):** We use $\mathrm{Beta}(1, 2)$ for the dispreferred group $\mathcal{G}^-$. This distribution has a mode at $(1-1)/(1+2-2) = 0$ and mean $1/3 \approx 0.33$, concentrating probability density near 0 to favor low consistency estimates.

The KL divergence $\mathrm{KL}(q \parallel p)$ measures the information loss when using prior p to approximate the posterior estimate q. By minimizing this divergence with the asymmetric priors described above, the estimator is regularized to produce values aligned with our domain knowledge: consistently high for preferred solutions and consistently low for dispreferred ones. This contrasts with a uniform $\mathrm{Beta}(1, 1)$ prior, which provides no directional guidance.

The weighting coefficient λ_{kl} controls the strength of this regularization and is typically set to 1.0 unless otherwise specified in ablation studies.

8.3 Prompt Templates and Generation Settings

We use the following decoding and sampling parameters throughout data generation:

```
"messages": messages,
"think_budget": 8192,
"max_tokens": 2048,
"temperature": 0.6,
"top_p": 0.95,
"top_k": 30,
"stream": False,
```

The first template is used in the main experiments to generate reverse reasoning problems.

```
"role_definition":
"You are an AI model tasked with generating a
reflective thinking exercise.
Given the following question and answer:"

- Question: {question}
- Answer: {answer}

"instructions":
"Your task is to reverse the roles of the
question and answer.
Transform the answer into a question that
is thought-provoking and encourages deeper
reflection.
Similarly, convert the original question into
a statement that serves as an insightful
answer.
Ensure that the new question remains
reasonable and stimulates further inquiry,
while the new answer is right to the question."

"expected_output":
- New Question:
- New Answer:
```

The second template elicits step-by-step reasoning and final answers from Qwen3-32B, serving as a basis for high-quality reasoning examples.

```
"role_definition":
"You are an AI model that is designed to
generate solutions to a given question.
All numerical answers must be explicitly
marked with \boxed{}."

- Question: {question}

"instructions":
"Ensure your answer is absolutely correct and
standard."

"expected_output":
Presents the complete and concise answer.
If the answer contains only one numerical
value, it must be marked in the form of
\boxed{}.
```

The third template is used to fact-check model-generated answers. It focuses on analytical verification, encouraging explicit reasoning and a concise binary verdict on correctness.

```
You are a meticulous fact-checking assistant.
1. Carefully reason through the model's answer
to the given question.
2. Use relevant knowledge, logical reasoning,
or explicit calculations to support your
analysis.
3. After reaching a conclusion, output exactly
two clean lines as follows:
- JUDGE: <yes|no>
('yes' if the model's verdict is factually
correct, 'no' otherwise.)
Question:
{question}
Model verdict (yes/no):
{model's answer}
```

The fourth template aims to construct reverse reasoning problems derived from verified forward examples.

```
You are an expert mathematical problem designer.
Given:
Original Problem:
{question}
Original Answer:
{model's answer}
Your task:
Create 3 reverse problems inspired by this original problem.
Each reverse problem must:
1. Be fully specified with no hidden or missing conditions.
2. Have exactly one unique correct answer, supported by clear reasoning for uniqueness.
3. Be meaningfully connected to the original problem by inverting knowns and unknowns, modifying parameters, or extending constraints.
Return four problems in the following structured format:
Problem 1
- Statement:
- Answer:
Problem 2
- Statement:
- Answer:
Problem 3
- Statement:
- Answer:
```

8.4 Multi-run Robustness of DGPO

Examples of Reverse Problem Construction To illustrate how reverse problems are constructed from original forward problems, we provide below one representative case.

Original problem: Find the arithmetic mean of all three-digit palindromes (numbers that read the same forward and backward

Original answer: 550.

Reverse problems:

1. *Given that the arithmetic mean of all three-digit palindromes is 550, find their total sum.*

2. *There are 90 three-digit palindromes in total. Find the remainder when the largest three-digit palindrome (999) is divided by this number.*

3. *How many three-digit palindromes cannot be expressed as the arithmetic mean of two other distinct three-digit palindromes?*

These examples demonstrate how reverse supervision is systematically constructed: each reverse problem maintains a close semantic link to the original forward problem while introducing a new perspective (e.g., altering the unknown, parameterizing radii, or changing tangency relations).

To assess stability, we repeat DGPO training three times for each base model family (Qwen3-1.7B-Base, SFT(Mixed), and Qwen3-1.7B), reporting mean accuracy and standard deviation across runs. Results are shown in Table 6. The small variance across runs suggests that DGPO yields consistent gains independent of random initialization and training noise.

8.5 Implication Details

We implement all experiments within the `SWIFT` framework (Zhao et al., 2024). Supervised fine-tuning (SFT) is performed on the Qwen3-1.7B-Base model using the standard `SWIFT` framework, trained for 3 epochs with a learning rate of 1×10^{-5}. DGPO is implemented by extending the framework with a custom `DGPO` module that augments the groupwise objective with posterior modulation. Training is performed for 2 epochs with a learning rate of 3×10^{-6} for Qwen3-1.7B-Base and 1×10^{-6} for Qwen3-1.7B, per-device batch size 1, and gradient accumulation 1 (global batch size of 4). All experiments employ DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) in bfloat16 precision for memory efficiency, running on 4×RTX A6000 (48GB) GPUs. For SFT, the maximum sequence length is set to 11,000 tokens to capture the full reasoning traces distilled from the teacher model. For DGPO training, we compare three initialization settings: (1) continuing from the SFT-distilled model, (2) training directly on Qwen3-1.7B-Base, and (3) training directly on Qwen3-1.7B. Depending on the dataset, training takes between one and four hours. In all DGPO cases, the maximum sequence length is capped at 1,000 tokens to balance long-form coverage with the computational feasibility of preference optimization.

8.6 Reverse Data Quality Details

Reverse Data Distillation. Table 5 reports the performance of SFT models trained on three reverse subsets constructed from bidirectional data. Across benchmarks, models distilled on reverse data achieve accuracy levels comparable to those obtained with the forward-only LIMO dataset, indicating that the generated reverse problems maintain a similar level of supervision quality. However, for RL-aligned backbones, performance declines con-

Table 5: Reverse Data Quality Evaluation.

Dataset	Base Model	AIME-25	GPQA	Math 500	GMQ	LMGH	Avg. Acc.
Subset 1	Qwen3-1.7B-Base	3.3%	29.8%	46.2%	33.6%	12.9%	25.2%
Subset 2	Qwen3-1.7B-Base	6.7%	29.3%	46.6%	33.6%	10.2%	25.3%
Subset 3	Qwen3-1.7B-Base	3.3%	28.3%	48.2%	35.3%	7.5%	24.5%
Subset 1	Qwen3-1.7B	6.7%	27.3%	34.8%	32.5%	5.2%	21.3%
Subset 2	Qwen3-1.7B	3.3%	25.3%	32.2%	36.6%	4.5%	20.4%
Subset 3	Qwen3-1.7B	6.7%	29.3%	40.2%	31.3%	5.3%	22.6%

Table 6: Statistical experiment information of DGPO averaged over three independent runs.

Base Model	AIME-25	GPQA	Math 500	GMQ	LMGH
Qwen3-1.7B-Base	$6.7\% \pm 11.1\%$	$28.6\% \pm 0.7\%$	$47.5\% \pm 1.0\%$	$35.2\% \pm 0.2\%$	$3.5\% \pm 0.5\%$
Qwen3-1.7B (Official)	$25.6\% \pm 4.7\%$	$31.6\% \pm 0.6\%$	$46.9\% \pm 1.6\%$	$36.3\% \pm 0.6\%$	$13.7\% \pm 2.8\%$

sistently across all benchmarks, suggesting that additional distillation on reverse data may interfere with previously optimized reward-aligned behaviors. These findings indicate that while reverse supervision offers strong groupwise learning signals that benefit base models, its advantages do not seamlessly extend to RL-aligned ones. Incorporating directional objectives into the reinforcement alignment process calls for further exploration to fully harness the potential of group-level reverse data.

8.7 Qualitative Comparison with DPO Variants

To further illustrate the difference between DGPO and pairwise preference baselines, we include the qualitative case study added during rebuttal. The task asks: *Find the sum of all integer bases* $b > 9$ *for which* 17_b *is a divisor of* 97_b.

DGPO answer chain. DGPO first converts the base-b expressions into their base-10 forms, then derives the divisibility condition that $b + 7$ must divide $9b+7$, which is equivalent to requiring $b+7$ to divide 56. It finally filters the valid divisors by the base constraint $b > 9$, subtracts 7 from each admissible divisor, and sums the resulting bases.

Vanilla DPO answer chain. Vanilla DPO identifies the divisibility condition and lists the divisors of 56 correctly, but it fails in the final filtering step. In particular, it retains invalid candidates such as 16, 25, and 36, which do not satisfy the full divisibility requirement, leading to an incorrect final set of bases.

Base-model answer chain. The unaligned Qwen3-1.7B-Base model identifies the initial condition and simplifies the expression, but it also keeps invalid bases that violate the constraint $b > 9$, ultimately producing an incorrect total of 275.

This example highlights the qualitative advantage of DGPO: the groupwise directional signal does not merely identify the right algebraic condition, but also helps the model preserve the final consistency check needed to rule out superficially plausible yet invalid solutions.