DiPO: Disentangled Perplexity Policy Optimization for Fine-grained Exploration-Exploitation Trade-Off

Xiaofan Li[1,2†], **Ming Yang**[2†], **Zhiyuan Ma**[2], **Shichao Ma**[2], **Jintao Du**[2], **Yu Cheng**[2], **Weiqiang Wang**[2], **Zhizhong Zhang**[1‡], **Xin Tan**[1], **Yanyun Qu**[4], **Lizhuang Ma**[1‡], **Yuan Xie**[1,3]
[1]East China Normal University, [2]Ant Group
[3]Shanghai Innovation Institute, [4]Xiamen University
lxfunzi@stu.ecnu.edu.cn, {lzma, yxie}@cs.ecnu.edu.cn

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has catalyzed significant advances in the reasoning capabilities of Large Language Models (LLMs). However, effectively managing the exploration and exploitation trade-off remains a critical challenge. In this paper, we fully analyze the exploration and exploitation dilemma of extremely hard and easy samples during the training and propose a new fine-grained trade-off mechanism. Concretely, we introduce a perplexity space disentangling strategy that divides the sample space into distinct exploration (high perplexity) and exploitation (low perplexity) subspaces, thereby mining fine-grained samples requiring exploration-exploitation trade-off. Subsequently, we propose a bidirectional reward allocation mechanism with a minimum impact on verification rewards to implement perplexity-guided exploration and exploitation, enabling more stable policy optimization. Finally, we have evaluated our method on two mainstream tasks: mathematical reasoning and function calling, and experimental results demonstrate the superiority of the proposed method, confirming its effectiveness in enhancing LLM performance by fine-grained exploration-exploitation trade-off. Code is available at https://github.com/FuNz-0/DiPO_RL

1 Introduction

Reinforcement Learning (RL) Ouyang et al. [2022], Shao et al. [2024], Yu et al. [2025] has emerged as a core technology in LLM post-training Comanici et al. [2025], Yang et al. [2025a], Team et al. [2026], Grattafiori et al. [2024], Lightman et al. [2023], driving LLMs from capability alignment to reasoning enhancement. Exploration-Exploitation Trade-Off (EETO) Auer et al. [2002] represents a core challenge in RL for LLM post-training, where error samples stand to benefit from further exploration, while correct samples are suitable for exploitation. In this paper, we propose the two dilemmas of EETO in GRPO-based methods. **First dilemma: the degradation of the advantage of extreme samples.** As illustrated in Figure 1 (a), a large proportions of sample groups fall into either the hard group (with uniform zero rewards) or the easy group (with uniform one rewards) during training. These extreme groups yield a zero advantage, leading to the lack of training gradients and thus exposes the GRPO paradigm to a dilemma of EETO, where the hard and easy groups lack signals for exploration and exploitation, respectively. **Second dilemma: ineffective exploration and exploitation.** As shown in Figure 1 (b), the distribution of perplexity (PPL) reflects the exploratory (high PPL) or exploitative (low PPL) tendency of samples, where some error samples exhibit an exploitative tendency, while some correct samples show an exploratory one. This ineffective exploration and exploitation severely decrease the stability and plasticity of RL.

[†]Equan Contribution
[‡]Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

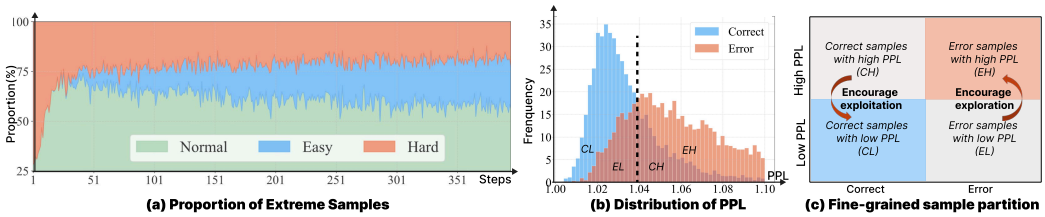


Figure 1: **(a)** The proportion of Easy/Normal/Hard groups in each step during the DAPO training. **(b)** The PPL distribution of correct and error samples in the validation set at 300th steps of DAPO training. **(c)** Illustration of four samples after PSD fine-grained partitioning.

Recent approaches have investigated the EETO by introducing PPL reward shaping. DACE Li et al. [2025] encourages all samples either to explore or to exploit based on intra-group accuracy. However, this coarse-grained partitioning would hinder policy optimization during early training stages when most samples belong to the hard group. Meanwhile, CDE Dai et al. [2025] designs multiple weighting mechanisms to employ exploration rewards at appropriate times, yet it does not explicitly address exploitation and introduces more hyper-parameters. Moreover, both CDE and DACE directly use perplexity as a reward bias, which may introduce additional uncertainty. In this paper, we propose a novel method **Di**sentangled Perplexity **P**olicy **O**ptimization (**DiPO**) to solve the two dilemmas and achieve a fine-grained EETO.

To enable fine-grained sample mining for exploratory and exploitative purposes, we propose the Perplexity Space Disentangling (PSD) strategy, that integrates statistical probability of correctness (verification reward) and PPL. In Figure 1 (b), error samples typically exhibit higher PPL, whereas correct samples generally maintain lower PPL. PSD infers the optimal threshold $\tau*$ by advantage judgment and minimizing classification errors, thus capturing the inherent correlation between PPL and sample correctness. Subsequently, as shown in Figure 1 (c), the entire sample space is partitioned into four fine-grained quadrants: correct samples with high PPL (CH), correct samples with low PPL (CL), error samples with high PPL (EH), and error samples with low PPL (EL). While the CL and EH samples naturally align with effective exploitation and exploration; The crux of EETO lies in encouraging exploitation for CH samples and exploration for EL samples.

In addition, the uncertainty of PPL distribution results in a significant discrepancy between the PPL reward and the verification reward. To achieve stable RL training for EETO, we abandon the direct adoption of PPL for reward shaping and proposes a Bidirectional Reward Reallocation (BRR) mechanism. Specifically, on the one hand, to avoid interfering with the policy optimization guided by the verification reward, BRR is only performed on the zero-gradient easy and hard groups. On the other hand, because the verification reward variance of the easy and hard groups is zero, we introduce a maximum-PPL reward reallocation strategy, where the rewards corresponding to the maximum PPL samples in the easy and hard groups are set to 0 and 1, respectively, ensuring the variance of the reallocated reward distribution is close to zero. The proposed BRR incorporates the PPL signal while minimizing the perturbation to the original reward distribution, thus realizing more stable training.

To summarize, the main contributions of this paper are:

- We propose a perplexity space disentanglement strategy that partitions the PPL space based on perplexity and correctness distributions, enabling a fine-grained EETO.
- We design a bidirectional reward reallocation mechanism that stabilizes training by reallocating rewards with minimal perturbation to the verification reward distribution."
- For mathematical reasoning and function calling tasks, our method achieves superior results confirming its effectiveness in reasoning enhancement for LLMs.

2 Preliminaries and Definitions

2.1 RLVR Algorithms

Group Relative Policy Optimization (GRPO) introduces a concise advantage calculation method. For each query $\boldsymbol{q}$ and its ground-truth answer $\boldsymbol{a}$, given the verifiable reward calculation function:

$$\mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a}) = 1 \quad \texttt{if is_equivalent}(\boldsymbol{a}, \boldsymbol{o}^i) \texttt{ else } \quad 0, \tag{1}$$

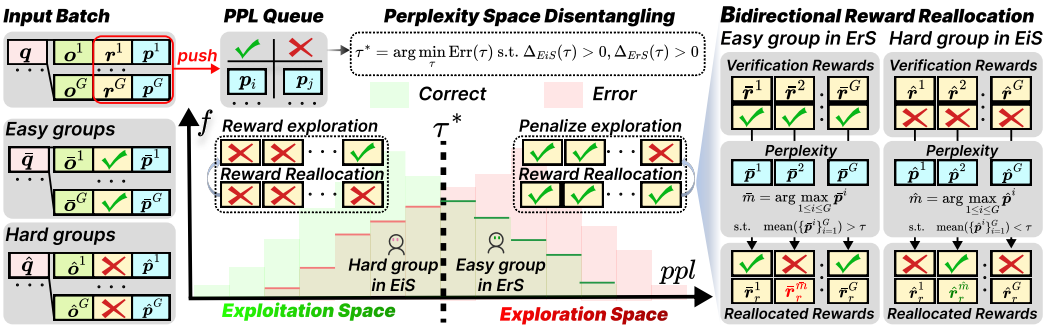


Figure 2: Illustration of DiPO, consisting of three modules: PPL Queue, Perplexity Space Disentangling (PSD), and Bidirectional Reward Reallocation (BRR). Specifically, the PPL Queue caches PPL items; PSD is used for fine-grained sample partition; and then BRR performs reward allocation.

where $\{\boldsymbol{o}^i\}_{i=1}^G$ is generated by rollout policy $\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{o}|\boldsymbol{q})$ through G-sampling. The estimated advantage $\hat{\boldsymbol{A}}_t^i$ under $\mathcal{R}$ is then computed as:

$$\hat{\boldsymbol{A}}_t^i(\mathcal{R}) = \frac{\mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a}) - \text{mean}(\{\mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a})\}_{i=1}^G)}{\text{std}(\{\mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a})\}_{i=1}^G)}. \tag{2}$$

When G-sampling rewards are all 0 or 1, all advantages come to 0, indicating the sampled group consists solely of hard groups and easy groups.

Dynamic Sampling Policy Optimization (DAPO) introduces new improvements based on GRPO The maximum optimization objective under $\mathcal{R}$ is as follows:

$$\mathcal{J}_{\text{DAPO}}(\theta, \mathcal{R}) = \mathbb{E}_{(\boldsymbol{q},\boldsymbol{a})\sim\mathcal{D},\{\boldsymbol{o}^i\}_{i=1}^G\sim\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{o}|\boldsymbol{q})} \frac{1}{\sum_{i=1}^G |\boldsymbol{o}^i|} \sum_{i=1}^G \sum_{t=1}^{|\boldsymbol{o}^i|} \left[\min\left(\boldsymbol{s}_t^i \hat{\boldsymbol{A}}_t^i(\mathcal{R}),\ \text{clip}\left(\boldsymbol{s}_t^i, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\hat{\boldsymbol{A}}_t^i(\mathcal{R})\right)\right],$$
$$\text{s.t.} \quad 0 < |\{\mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a}) | \mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a}) = 1)\}| < G \tag{3}$$

where $\boldsymbol{s}_t^i = \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{o}_t^i|\boldsymbol{q},\boldsymbol{o}_{<t}^i)}{\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{o}_t^i|\boldsymbol{q},\boldsymbol{o}_{<t}^i)}$ is the ratio of importance sampling. In this paper, we use advanced DAPO as our baseline without overlong reward shaping, as in some cases, overlong reward shaping would damage the model's performance.

2.2 Perplexity

Perplexity (shorted as PPL) in Large Language Models is a metric that quantifies how confidently a probability model predicts a sample. Mathematically, given a model $\pi_{\boldsymbol{\theta}}$, a question $\boldsymbol{q}$, and a response $\boldsymbol{o}^i = (\boldsymbol{o}_1^i, \boldsymbol{o}_2^i, ..., \boldsymbol{o}_T^i)$ generated by $\pi_{\boldsymbol{\theta}}$, the perplexity is given by:

$$\boldsymbol{p}^i = \exp\left(-\frac{1}{T}\sum_{t=1}^T \log \pi_\theta(\boldsymbol{o}_t^i|\boldsymbol{q}, \boldsymbol{o}_{<t})\right) \tag{4}$$

For a response, a low PPL indicates that the trajectory tends to *exploitation*, while higher PPL indicates that the trajectory tends to *exploration* Dai et al. [2025], Sun et al. [2025].

3 Methodology

3.1 Overview

As illustrated in Figure 2, the proposed DiPO mainly contains two modules: 1) Perplexity Space Disentangling (PSD); 2) Bidirectional reward reallocation (BRR). PSD calculates conditional probability distribution based on the PPL Queue cached with PPL-reward pairs, and then determines the optimal threshold by the advantage judgment and minimizing classification errors. We term higher and lower PPL region exploration space (ErS) and exploitation space (EiS), respectively. Through PPL disentanglement, the ineffective samples for exploration and exploitation, i.e., hard groups in

EiS and Easy groups in ErS are selected for policy optimization. Subsequently, BRR is implemented for these samples; for the hard group in EiS, the reward for maximum PPL sample is set to 1, thus encouraging exploration; the easy group in ErS, the reward for maximum PPL sample is set to zero, thus penalizing exploration. BRR incorporates the PPL signal and minimizes perturbations to the verification reward, enabling more stable training.

3.2 Perplexity Space Disentangling

PSD calculates the optimal threshold τ^* for dividing EiS and ErS by introducing the conditional probability distribution $\Pr(R|P)$ where P and R represent PPL and reward. For EETO, error samples stand to benefit from further exploration, while correct samples are suitable for exploitation. Thus the optimal EiS and ErS should have the highest relevance to sample correctness, where samples falls in EiS are considered more likely to be correct, whereas those in ErS are generally deemed error.

Online Statistical Estimation. Since the PPL distribution induced by the policy changes dynamically during training, direct estimation of conditional probabilities may suffer from an unfavorable variance-bias trade-off. To stabilize the estimation, we maintain a PPL queue $\mathcal{Q}$ that caches samples from the most recent two batches: $\mathcal{Q} = \{(\boldsymbol{p}_i, \boldsymbol{r}_i)\}_{i=1}^{M}$, where $\boldsymbol{p}_i \in [1, +\infty)$ denotes the perplexity (PPL) of the i-th sample and $\boldsymbol{r}_i \in \{0, 1\}$ denotes its corresponding validation reward.

For a given threshold τ, we estimate the empirical conditional probabilities of the reward given whether the PPL is below or above τ as

$$\begin{aligned}
\widehat{\Pr}(R=1 \mid P<\tau) &= \frac{\sum_{i=1}^{M} \mathbb{I}(p_i<\tau,\, r_i=1)}{\sum_{i=1}^{M} \mathbb{I}(p_i<\tau)}, \quad \widehat{\Pr}(R=0 \mid P<\tau) = \frac{\sum_{i=1}^{M} \mathbb{I}(p_i<\tau,\, r_i=0)}{\sum_{i=1}^{M} \mathbb{I}(p_i<\tau)}, \\
\widehat{\Pr}(R=1 \mid P>\tau) &= \frac{\sum_{i=1}^{M} \mathbb{I}(p_i>\tau,\, r_i=1)}{\sum_{i=1}^{M} \mathbb{I}(p_i>\tau)}, \quad \widehat{\Pr}(R=0 \mid P>\tau) = \frac{\sum_{i=1}^{M} \mathbb{I}(p_i>\tau,\, r_i=0)}{\sum_{i=1}^{M} \mathbb{I}(p_i>\tau)}.
\end{aligned} \tag{5}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. M denotes the size of PPL queue $\mathcal{Q}$.

To quantify the statistical uncertainty of each estimate, we compute 95% confidence intervals using the Normal (Wald) approximation Wald [2004]. For a generic empirical probability estimate $\hat{p} = m/n$, where m denotes the number of successes among n samples, the standard error is $\mathrm{SE}(\hat{p}) = \sqrt{\hat{p}(1-\hat{p})/n}$, and the corresponding 95% confidence interval is

$$[\, \hat{p} - 1.96 \cdot \mathrm{SE}(\hat{p}),\ \hat{p} + 1.96 \cdot \mathrm{SE}(\hat{p}) \,]\,. \tag{6}$$

These estimated probabilities represent the sample likelihood of being correct or error in the *Exploitation Space* (EiS, $P < \tau$) and *Exploration Space* (ErS, $P > \tau$) demarcated by the threshold τ.

Advantage Judgment. During the early RL stage of pre-trained models, the correlation between PPL and correctness may not be positively correlated (see Appendix E.8), where directly activating PSD would hinder optimization. To this end, we introduce a advantage judgment mechanism based on the results of online statistical estimation.

Specifically, for a given threshold τ, we define two advantage functions that characterize the separation between correct samples and error samples in divided PPL regions:

$$\begin{aligned}
\Delta_{\mathrm{EiS}}(\tau) &= \Pr(R=1 \mid P<\tau) - \Pr(R=0 \mid P<\tau), \\
\Delta_{\mathrm{ErS}}(\tau) &= \Pr(R=0 \mid P>\tau) - \Pr(R=1 \mid P>\tau),
\end{aligned} \tag{7}$$

where $\Delta_{\mathrm{EiS}}(\tau)$ measures the *correctness advantage* in the EiS, and $\Delta_{\mathrm{ErS}}(\tau)$ measures the *error advantage* in the ErS.

To reduce the impact of sampling randomness and ensure robust judgment, we conservatively evaluate these gaps using the boundary values of the 95% confidence intervals derived in the online estimation stage (Equation (6)). Concretely, the two judgment functions are as follows:

$$\Delta_{\mathrm{EiS}}(\tau) = L_1 - U_2 \quad \text{and} \quad \Delta_{\mathrm{ErS}}(\tau) = L_4 - U_3, \tag{8}$$

where $[L_1, U_1]$ and $[L_2, U_2]$ denote the 95% confidence intervals of $\Pr(R = 1 \mid P < \tau)$ and $\Pr(R = 0 \mid P < \tau)$, respectively, $[L_3, U_3]$, $[L_4, U_4]$ correspond to $\Pr(R = 1 \mid P > \tau)$ and $\Pr(R = 0 \mid P > \tau)$. Considering the PPL space to be meaningfully correlated with correctness at threshold τ only when both conditions $\Delta_{\mathrm{EiS}}(\tau) > 0$ and $\Delta_{\mathrm{ErS}}(\tau) > 0$ hold.

Minimizing Classification Errors. In the middle and later RL stages, a wide range of thresholds may satisfy Advantage Judgment, necessitating the further selection of an optimal value. We formalize a classification task using PPL as the criterion: responses with a PPL below τ are classified as correct, while those above are deemed error. By minimizing classification errors (Err(τ) in Figure 2) to find the optimal threshold:

$$\begin{aligned} \tau* = \arg\min_{\tau} \frac{1}{|\mathcal{Q}|} \sum_{(\boldsymbol{r}_i, \boldsymbol{p}_i) \in \mathcal{Q}} |\boldsymbol{r}_i - \mathbb{I}(\boldsymbol{p}_i < \tau)| \\ \text{s.t. } \Delta_{EiS}(\tau) > 0, \ \ \Delta_{ErS}(\tau) > 0 \end{aligned} \tag{9}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Through advantage judgment and minimizing classification errors, the disentangled PPL spaces are highly correlated with correctness. Consequently, the hard groups in EiS and the easy groups in ErS would be the important samples for achieving the EETO. Algorithm details are illustrated in Algorithm 1.

3.3 Bidirectional Reward Reallocation

To implement a stable exploration-exploitation optimization and minimize the impact to verification rewards, BRR does not directly employ PPL as the basis for reward shaping, but introduces the maximum-PPL reward reallocation strategy. BRR thus aims to drive updates toward the exploratory (high-entropy) direction for the hard groups in EiS, and toward the exploitative (low-entropy) direction for the easy groups in ErS.

Proposition 1 (Entropy Increase with Maximum-PPL Reward)**.** *Let π_t be the policy at step t. Given a query $\boldsymbol{q}$ and a group of outputs $\{\boldsymbol{o}^i\} \sim \pi_t(\cdot \mid \boldsymbol{q})$, if the output with the maximum PPL among $\{\boldsymbol{o}^i\}$ is assigned a reward, then the average entropy of the updated policy $\pi_{t+1}(\cdot \mid \boldsymbol{q})$ tends to increase.*

Reward Reallocation for Hard Groups. As shown in Figure 2, we define a hard group as $\hat{\boldsymbol{q}}$, $\{\hat{\boldsymbol{o}}^i\}$, $\{\hat{\boldsymbol{r}}^i\}$, $\{\hat{\boldsymbol{p}}^i\}$, here $\{\hat{\boldsymbol{r}}^i\} = \{0\}$. If $mean(\{\hat{\boldsymbol{p}}^i\}) < \tau*$ means that this hard group is located in EiS, the goal of the reward reallocation is to update it towards the high entropy direction. Concretely, the index $\hat{m}$ corresponding to the maximum PPL in $\{\hat{\boldsymbol{o}}^i\}$ is identified: $\hat{m} = \arg\max_{1 \le i \le G} \hat{\boldsymbol{p}}^i$. Subsequently, the reallocated reward $\{\hat{\boldsymbol{r}}_r^i\}$ is constructed by setting $\hat{\boldsymbol{r}}_r^m = 1$ while preserving all other original values, such that $\hat{\boldsymbol{r}}_r^i = \hat{\boldsymbol{r}}^i$ for all $i \neq \hat{m}$. Noted that if $\tau*$ does not exist, or $mean(\{\hat{\boldsymbol{p}}^i\}) > \tau*$, reallocated reward remains unchanged, i.e., $\{\hat{\boldsymbol{r}}_r^i\} = \{\hat{\boldsymbol{r}}^i\} = \{0\}$.

Proposition 2 (Entropy decrease with Maximum-PPL penalty)**.** *Let π_t be the policy at step t. Given a query $\boldsymbol{q}$ and a group of outputs $\{\boldsymbol{o}^i\} \sim \pi_t(\cdot \mid \boldsymbol{q})$, if the output with the maximum PPL among $\{\boldsymbol{o}^i\}$ is assigned a penalty, then the average entropy of the updated policy $\pi_{t+1}(\cdot \mid \boldsymbol{q})$ tend to decrease.*

Reward Reallocation for Easy Groups. As shown in Figure 2, reward reallocation for the easy group is the opposite of that for the hard group. A easy group is defined as $\bar{\boldsymbol{q}}$, $\{\bar{\boldsymbol{o}}^i\}$, $\{\bar{\boldsymbol{r}}^i\}$, $\{\bar{\boldsymbol{p}}^i\}$, here $\{\bar{\boldsymbol{r}}^i\} = \{1\}$. If $mean(\{\bar{\boldsymbol{p}}^i\}) > \tau*$ means that this easy group is located in ErS, the goal of the reward reallocation is to update it towards the low entropy direction. Concretely, the index $\bar{m}$ corresponding to the maximum PPL in $\{\bar{\boldsymbol{o}}^i\}$ is identified: $\bar{m} = \arg\max_{1 \le i \le G} \bar{\boldsymbol{p}}^i$. Subsequently, the reallocated reward $\{\bar{\boldsymbol{r}}_r^i\}$ is constructed by setting $\bar{\boldsymbol{r}}_r^m = 0$ while preserving all other original values, such that $\bar{\boldsymbol{r}}_r^i = \bar{\boldsymbol{r}}^i$ for all $i \neq \bar{m}$. Noted that if $\tau*$ does not exist, or $mean(\{\bar{\boldsymbol{p}}^i\}) < \tau*$, reallocated reward remains unchanged, i.e., $\{\bar{\boldsymbol{r}}_r^i\} = \{\bar{\boldsymbol{r}}^i\} = \{1\}$.

Finally, we abstracted the process of BRR into a formalized reward function $\mathcal{R}_r$, defined as follows:

$$\mathcal{R}_r(\{\boldsymbol{r}^i\}) = \begin{cases} \{\hat{\boldsymbol{r}}_r^i\} & \text{if } \{\boldsymbol{r}^i\} = \{0\}, \\ \{\bar{\boldsymbol{r}}_r^i\} & \text{if } \{\boldsymbol{r}^i\} = \{1\}, \\ \{0\} & \text{other}, \end{cases} \tag{10}$$

where BRR only performs reward reallocation for the easy groups and hard groups, and sets all rewards of the normal groups to zero, allowing the objectives associated with the reallocated rewards (denoted as $\mathcal{R}_r$) and verification rewards (denoted as $\mathcal{R}$) to be weighted separately. Algorithm details are illustrated in Algorithm 2.

Policy Optimization. The objectives associated with $\mathcal{R}$ and $\mathcal{R}_r$ can be weighted separately. Thus, we use a hyper-parameter α to control the loss weight of $\mathcal{R}_r$. The maximum optimization objective is as follows:

$$\mathcal{J}_{DiPO}(\theta) = \mathcal{J}_{DAPO}(\theta, \mathcal{R}) + \alpha \times \mathcal{J}_{DAPO}(\theta, \mathcal{R}_r), \tag{11}$$

Table 1: Comparison of mathematical reasoning in ACC/mean@8 on 6 mathematics benchmarks. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

Method	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base							
Base model	9.58	3.75	48.48	31.48	25.52	24.82	23.94
GRPO	<u>26.67</u>	23.33	85.83	60.24	53.06	44.39	48.92
DAPO	26.25	23.75	86.43	61.90	53.88	44.34	49.43
DAPO w/ EL	<u>26.67</u>	**24.58**	<u>86.78</u>	<u>62.95</u>	**54.53**	<u>44.53</u>	<u>50.01</u>
CDE	<u>26.67</u>	<u>24.17</u>	85.93	62.35	52.25	43.11	49.08
DiPO (ours)	**29.17**	**24.58**	**87.00**	**63.70**	<u>54.09</u>	**44.76**	**50.55**
Qwen3-8B-Base							
Base model	8.33	9.17	66.78	39.46	33.30	31.75	31.47
GRPO	31.67	24.58	89.08	69.28	56.20	**48.62**	53.24
DAPO	30.08	25.83	89.43	69.12	56.90	<u>48.02</u>	53.23
DAPO w/ EL	<u>33.75</u>	25.42	**89.58**	<u>69.87</u>	<u>57.21</u>	47.56	<u>53.90</u>
CDE	31.67	<u>26.25</u>	89.35	68.07	57.11	47.75	53.37
DiPO (ours)	**35.00**	**27.50**	<u>89.55</u>	**71.23**	**57.73**	47.75	**54.79**
Qwen2.5-7B							
Base model	7.08	2.08	41.53	22.74	19.28	17.19	18.32
GRPO	<u>20.42</u>	<u>15.42</u>	79.15	58.43	42.42	36.95	42.13
DAPO	<u>20.42</u>	**16.67**	79.08	<u>59.94</u>	42.70	37.55	<u>42.73</u>
DAPO w/ EL	20.00	14.58	<u>79.85</u>	58.73	<u>43.05</u>	**39.65**	42.64
CDE	20.00	15.00	79.00	55.87	42.94	35.94	41.46
DiPO (ours)	**22.92**	**16.67**	**80.35**	**60.09**	**43.72**	<u>37.59</u>	**43.56**

where $\mathcal{J}_{DAPO}$ is the DAPO optimization objective in Eq. (3). According to Propositions 1 and 2, hard groups in EiS will have an increased entropy, tending to explore, while easy groups in ErS will have a decreased entropy, tending to exploit, achieving EETO in policy updating.

4 Experiment

4.1 Experiment Setup

To comprehensively evaluate the effectiveness of our proposed DiPO, we conduct experiments on two downstream tasks of LLMs: mathematical reasoning and function calling. Detailed configurations are provided below. Additional analyses of training-run robustness, perplexity queue size, and PSD activation and threshold dynamics are provided in Appendix Sections E.1, E.2, and E.3, respectively.

Mathematical Reasoning. We evaluate DiPO on the mathematical reasoning task using the DAPO-17K Yu et al. [2025] dataset for training. Model performance is assessed on six challenging benchmarks: AIME24, AIME25, AMC23 (denoted as AMC), MATH500 Hendrycks et al. [2021] (denoted as MATH) the OE_TO_mat_en_COMP subset from OlympiadBench He et al. [2024](denoted as OLY), and Minerva (denoted as MIN). These evaluations are designed to comprehensively examine the method's effectiveness across diverse problem difficulties and formats.

We perform a comparative analysis against several strong baselines: 1. GRPO Shao et al. [2024], 2. DAPO Yu et al. [2025], 3. DAPO enhanced with Entropy Loss (DAPO w/ EL) Williams [1992], and 4. CDE Dai et al. [2025] with PPL reward shaping. All mathematical reasoning experiments are implemented using the VERL Sheng et al. [2025] framework. To ensure a rigorous comparison, we employ three different pretrained base models: Qwen3-4B-Base, Qwen3-8B-Base Yang et al. [2025a], and Qwen2.5-7B Yang et al. [2025b]. All models utilize consistent training configurations and prompts, as detailed in Appendix D.1.

Function Calling. We establish our experimental setup using the publicly available ToolRL Qian et al. [2025] framework as the primary baseline. Furthermore, we implement ToolRL augmented

Table 2: Comparison of function calling in acc on BFCLv3 benchmark. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

Method	Non-Live Acc	Live Acc	Multi Turn Acc	Relevance Detection	Irrelevance Detection
Qwen2.5-3B-Instruct					
Base model	42.52	53.96	1.00	64.71	56.01
SFT400	69.29	41.40	0.00	<u>94.44</u>	8.11
SFT400+PPO	78.29	58.76	5.12	94.12	54.70
SFT400+GRPO	76.21	64.15	1.75	88.89	71.76
PPO, Cold Start	<u>82.42</u>	67.78	4.88	94.12	<u>73.87</u>
ToolRL+GRPO	81.58	**73.78**	3.75	88.24	**84.85**
ToolRL+DAPO	82.19	69.43	<u>8.00</u>	81.25	57.60
ToolRL+DiPO	**83.42**	<u>73.06</u>	**8.62**	**100.00**	54.16
Qwen2.5-7B-Instruct					
Base model	66.02	53.51	4.25	76.47	62.66
SFT400	69.29	41.40	0.00	<u>94.44</u>	8.11
SFT400+PPO	83.90	51.84	0.25	**100.00**	29.66
SFT400+GRPO	80.69	46.51	0.25	**100.00**	14.19
PPO, Cold Start	79.33	63.17	0.38	88.89	52.92
ToolRL+GRPO	86.17	74.90	18.12	83.33	**76.68**
ToolRL+DAPO	**87.10**	<u>76.31</u>	<u>19.75</u>	87.50	67.25
ToolRL+DiPO	<u>86.21</u>	**76.83**	**24.50**	87.50	<u>69.57</u>

with DAPO (ToolRL+DAPO) for comparison. Since ToolRL does not utilize a binary verification reward, certain modifications were made when implementing TooRL+DiPO.

We conduct comprehensive testing on the BFCLv3 benchmark Patil et al. [2025], which provides a diverse set of function calling scenarios for thorough evaluation. Consistent with the TooRL setup, we use the same model architectures (Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct) and maintain identical training configurations and datasets across all compared methods, details in Appendix D.2.

4.2 Comparison Results of Mathematical Reasoning

Table 1 shows the comprehensive comparison results of mathematical reasoning in ACC/mean@8 across 6 benchmarks evaluated on three base models, our proposed DiPO demonstrates consistent and superior performance enhancements over other reinforcement learning algorithms. Overall, DiPO achieves the highest average score (AVG) across all three model scales, with 50.55% for Qwen3-4B-Base, 54.79% for Qwen3-8B-Base, and 43.56% for Qwen2.5-7B, surpassing all compared baselines. Specifically, on the more challenging AIME benchmarks, DiPO attains the best results in most cases, such as 29.17% on AIME24 and 24.58% on AIME25 with Qwen3-4B-Base, and notably 35.00% on AIME24 and 27.50% on AIME25 with Qwen3-8B-Base, indicating its strong capability in handling complex mathematical reasoning tasks. Furthermore, the performance gains are particularly pronounced in larger models like Qwen3-8B-Base, where DiPO outperforms the second-best method by 0.89 percentage points in AVG, highlighting its scalability and effectiveness in leveraging model capacity. Notably, although DAPO with EL achieves highly competitive results, it demonstrates pronounced sensitivity to its configuration coefficients, as further examined in the Appendix E.6.

4.3 Comparison Results of Function Calling

Building upon the mathematical reasoning evaluation, we further assess the performance of DiPO on the function calling task. As presented in Table 2, DiPO achieves the highest Multi-Turn Acc on both model scales, outperforming ToolRL+DAPO by 0.62 and 4.75 percentage points on Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, respectively. These component-level results support the effectiveness of DiPO in multi-turn function calling.

Table 3: The impact of hyperparameter α on performance. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

α	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base							
$\alpha = 0.0$	<u>26.25</u>	23.75	<u>86.43</u>	61.90	53.88	44.34	49.43
$\alpha = 0.1$	**29.17**	**24.58**	**87.00**	**63.70**	**54.09**	<u>44.76</u>	**50.55**
$\alpha = 1.0$	25.83	<u>24.17</u>	86.38	<u>62.05</u>	<u>54.01</u>	**45.50**	<u>49.66</u>
Qwen3-8B-Base							
$\alpha = 0.0$	30.08	<u>25.83</u>	<u>89.43</u>	69.12	<u>56.90</u>	<u>48.02</u>	53.23
$\alpha = 0.1$	**35.00**	**27.50**	**89.55**	**71.23**	**57.73**	47.75	**54.79**
$\alpha = 1.0$	<u>32.08</u>	24.58	88.68	<u>70.03</u>	56.71	**48.07**	<u>53.36</u>

4.4 Hyperparameter Analysis

The orthogonal design between the reallocated rewards and validation rewards enables precise, independent adjustment of DiPO's influence via the hyperparameter α. Table 3 presents the performance variations of Qwen3-4B-Base and Qwen3-8B-Base models under different values of α. Setting $\alpha = 0.1$ achieves the highest AVG on both model scales, improving over $\alpha = 0.0$ by 1.12 and 1.56 percentage points on the 4B and 8B models, respectively. Setting $\alpha = 0.0$ removes the contribution of BRR.

4.5 Ablation Experiment and Discussion

Table 4 presents ablation results of the contributions of PSD and BRR on Qwen3-4B-Base and Qwen3-8B-Base on six mathematical reasoning benchmarks. PPL reward refers to using PPL directly as rewards: positive PPL for hard groups, and negative PPL for easy groups. Overall, the combination of PSD and BRR achieves the best performance for both 4B and 8B models, verifying their effectiveness. Detailed discussions are as follows.

Discussion 1. Fine-grained exploration and exploitation make RL more effective. In Table 4, without PSD, all hard samples are indiscriminately driven toward high PPL, whereas easy samples are steered toward low PPL. PSD improves this by disentangling the space into ErS (high PPL) and EiS (low PPL), allowing for fine-grained optimization: easy groups within ErS are encouraged toward EiS, while hard groups within EiS are directed toward ErS. The results confirm the critical role of PSD in model enhancement. Comparing PSD + PPL reward with PPL reward alone, adding PSD improves AVG by 2.07 points on the 4B model and 1.70 points on the 8B model. Comparing PSD + BRR with BRR alone, the corresponding gains are 3.32 points and 3.42 points, respectively. Significant improvements indicate that invalid exploration and exploitation are needed for optimization, while excessive encouragement of exploration and exploitation is detrimental.

Discussion 2. Reward shaping should not cause significant changes to the intrinsic validation rewards. In Table 4, we compare our BRR against PPL reward. The experimental results confirm that BRR is superior to the PPL reward method. Concretely, with using PSD, BRR compared to PPL reward brings an improvement of 0.93 and 2.23 respectively on the 4B and 8B models. Compared with the baseline, PSD + PPL reward improves AVG by 0.19 points on the 4B model, but reduces AVG by 0.67 points on the 8B model. Science the reward variance of easy and hard groups are both zeros, directly using PPL as a reward causes drastic shifts in the reward distribution, potentially destabilizing the training process. In contrast, BRR introduces the reallocated rewards maintain a distribution very similar to the original validation rewards, which allows BRR to leverage PPL signals for enhancement with the minimum impact on the validation reward landscape.

4.6 Quantitative Results

To further verify the properties of DiPO, we conducted a quantitative analysis of some data during the training process of DAPO and DiPO.

Exploration-exploitation trade-off. Figure 3 shows the PPL distribution of the Qwen3-8B-Base after training on DAPO-17K with DAPO and DiPO. The results demonstrate that under the DAPO

Table 4: Ablation study results on Qwen3-4B-Base and Qwen-8B-Base, showing ACC/mean@8 on six mathematical reasoning benchmarks. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

PSD	BRR	PPL reward	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base									
✗	✗	✗	26.25	23.75	86.43	61.90	53.88	44.34	49.43
✗	✓	✗	25.83	20.83	85.10	59.19	49.37	43.06	47.23
✗	✗	✓	24.58	23.75	84.60	61.45	49.09	41.82	47.55
✓	✗	✓	27.50	23.33	86.35	62.35	54.00	44.21	49.62
✓	✓	✗	**29.17**	**24.58**	**87.00**	**63.70**	**54.09**	**44.76**	**50.55**
Qwen3-8B-Base									
✗	✗	✗	30.08	25.83	89.43	69.12	56.90	**48.02**	53.23
✗	✓	✗	30.08	24.58	88.35	65.36	53.62	46.23	51.37
✗	✗	✓	27.92	25.42	88.00	64.76	53.45	45.63	50.86
✓	✗	✓	30.83	24.58	88.15	68.83	56.65	46.30	52.56
✓	✓	✗	**35.00**	**27.50**	**89.55**	**71.23**	**57.73**	47.75	**54.79**

training, the PPL distribution of correct and incorrect samples exhibits a significant overlap. This phenomenon directly causes a large number of hard samples to lose exploration capability, thereby limiting the plasticity of RL. In contrast, DiPO shows superior distribution characteristics: error samples are more located in the high-PPL region, while correct samples are still concentrated in the low-PPL region, which achieves a balanced trade-off between exploration and exploitation in RL.

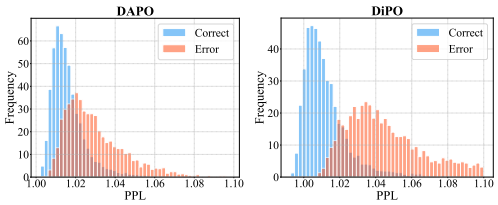


Figure 3: PPL distribution comparison of correct and error samples for Qwen3-8B-Base trained on DAPO-17K Dataset via DAPO and DiPO.

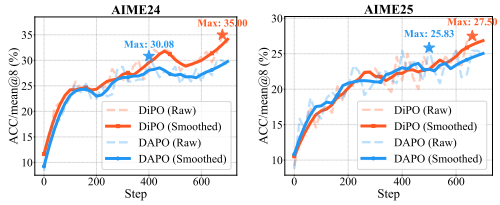


Figure 4: ACC/mean@8 curves of DiPO and DAPO (raw and smoothed curves) on AIME24 and AIME25 with using Qwen3-8B-Base model.

Higher upper bound for later training. Figure 4 presents the test curves of DiPO and DAPO on two AIME benchmarks during the training process. The results show that at the initial stage of training, there was no significant difference in the performance of the two methods; in the later stages of training, however, the performance curve of DAPO exhibited a significant slowdown in growth, whereas DiPO maintained a sustained capacity for exploratory improvement. Combined with the PPL distribution characteristics in Figure 4, this phenomenon can be further explained: DiPO enables hard samples to conduct more sufficient exploration, while the training paradigm of DAPO leads to a conservative tendency in the exploration strategy of hard samples.

5 Conclusion

This paper explores the exploration-exploitation trade-off between in RL training. We analyze the two EE dilemmas faced by the extreme hard and easy groups, and proposed the Disentangled Perplexity Policy Optimization. First, we develop a novel Perplexity Space Disentangling method to identify which hard samples require encouragement for exploration and which easy samples need promotion for exploitation. Then, we design a Bidirectional Reward Reallocation mechanism that incorporates PPL-based exploration and exploitation signals while minimizing disruptions to the original verification reward distribution. Finally, extensive experiments on mathematical reasoning and function calling tasks demonstrate the comprehensive superiority of the proposed method.

6 Acknowledgments

This work is supported by the Science and Technology Commission of Shanghai Municipality (Grant No. 25511102700), National Natural Science Foundation of China (Grant No. 72192820, 62476090, 62302167, U23A20343, 62502159), Natural Science Foundation of Shanghai (Grant No. 25ZR1402135), Natural Science Foundation of Chongqing (Grant No. CSTB2023NSCQ-JQX0007, CSTB2025NSCQ-GPX0445), Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2501), Zhejiang University, Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science-MOE, ECNU.

References

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Liang Chen, Xueting Han, Qizhou Wang, Bo Han, Jing Bai, Hinrich Schutze, and Kam-Fai Wong. EEPO: exploration-enhanced policy optimization via sample-then-forget. *arXiv preprint arXiv:2510.05837*, 2025a.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025b.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Runpeng Dai, Linfeng Song, Haolin Liu, Zhenwen Liang, Dian Yu, Haitao Mi, Zhaopeng Tu, Rui Liu, Tong Zheng, Hongtu Zhu, et al. Cde: Curiosity-driven exploration for efficient reinforcement learning in large language models. *arXiv preprint arXiv:2509.09675*, 2025.

Jia Deng, Jie Chen, Zhipeng Chen, Daixuan Cheng, Fei Bai, Beichen Zhang, Yinqian Min, Yanzipeng Gao, Wayne Xin Zhao, and Ji-Rong Wen. From trial-and-error to improvement: A systematic analysis of llm exploration mechanisms in rlvr. *arXiv preprint arXiv:2508.07534*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

Haoyang He, Zihua Rong, Kun Ji, Chenyang Li, Qing Huang, Chong Xia, Lan Yang, and Honggang Zhang. Rethinking reasoning quality in large language models through enhanced chain-of-thought via rl, 2025. *URL https://arxiv. org/abs/2509.06024*, 2025.

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

Ang Li, Zhihang Yuan, Yang Zhang, Shouda Liu, and Yisen Wang. Know when to explore: Difficulty-aware certainty as a guide for llm reinforcement learning. *arXiv preprint arXiv:2509.00125*, 2025.

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

Jia Liu, ChangYi He, YingQiao Lin, MingMin Yang, FeiYang Shen, and ShaoGuo Liu. Ettrl: Balancing exploration and exploitation in llm test-time reinforcement learning via entropy mechanism. *arXiv preprint arXiv:2508.11356*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*, 2025.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.

Yan Sun, Jia Guo, Stanley Kok, Zihao Wang, Zujie Wen, and Zhiqiang Zhang. Efficient reinforcement learning for large language models with intrinsic exploration. *arXiv preprint arXiv:2511.00794*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Chenxiao Gao, et al. Kimi k2: Open agentic intelligence, 2026. URL `https://arxiv.org/abs/2507.20534`.

Abraham Wald. *Sequential analysis*. Courier Corporation, 2004.

Christian Walder and Deep Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. *arXiv preprint arXiv:2505.15201*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Rihui Xin, Han Liu, Zecheng Wang, Yupeng Zhang, Dianbo Sui, Xiaolin Hu, and Bingning Wang. Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *arXiv preprint arXiv:2505.19439*, 2025.

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

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, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025b. URL `https://arxiv.org/abs/2412.15115`.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, et al. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning. *arXiv preprint arXiv:2508.16949*, 2025.

A Proof of Section 3.3

A.1 First-Order Entropy Analysis

Proof. Given a language model π_θ with parameters θ. Given a query q, sample G responses $\{o^i\}_{i=1}^G$. Each response o^i is a token sequence $(o_1^i, \ldots, o_{T^i}^i)$. The model's probability for token o_t^i given context $(q, o_{<t}^i) = \pi_\theta(y \mid q, o_{<t}^i)$ is denoted as $\pi_t^i(o_t^i)$. The probability of other tokens in position of the t-th token is denoted as $\pi_t^i(y)$, and the logistic of y is denoted as $l_i^t(y)$, $y \in \mathcal{V}$ and $\mathcal{V}$ is the vocabulary of π_θ. For convenience, we do not consider the clip operation, the loss function is as follows:

$$L(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o^i|}\sum_{t=1}^{|o^i|}\hat{A}_t^i\frac{\pi_t^i(o_t^i)}{\pi_{old}(o_t^i)}, \text{ where } \quad \pi_t^i(o_t^i) = \frac{exp(l_i^t(o_i^t))}{\sum_{y\in\mathcal{V}} exp(l_i^t(y))}. \tag{12}$$

Optimization gradient descent is $\theta' = \theta - \eta\nabla_\theta L$, with small learning rate η. Define the token-level entropy for response i at token t:

$$H_t^i = -\sum_{y\in\mathcal{V}}\pi_\theta(y \mid q, o_{<t}^i)\log\pi_\theta(y \mid q, o_{<t}^i) = -\sum_{y\in\mathcal{V}}\pi_t^i(y)\log\pi_t^i(y). \tag{13}$$

The average token entropy over all responses and time steps is:

$$H_{\text{avg}} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o^i|}\sum_{t=1}^{|o^i|}H_t^i. \tag{14}$$

The following will prove that high PPL rewards and high PPL penalties can respectively increase and decrease H_{avg}.

Logit Change. Consider a single token of response o_t^i for training, the loss contribution is $-\hat{A}_t^i\frac{\pi_t^i(o_t^i)}{\pi_{old}(o_t^i)}$. The gradient of the loss with respect to logit $l_t^i(y)$ is:

$$\frac{\partial(-\hat{A}_t^i\frac{\pi_t^i(y)}{\pi_{old}(y)})}{\partial l_t^i(y)} = \hat{A}_t^i\frac{\pi_t^i(y)}{\pi_{old}(y)}(\pi^i(y) - \mathbf{1}_{y=o_t^i}) \approx \hat{A}_t^i(\pi^i(y) - \mathbf{1}_{y=o_t^i}). \tag{15}$$

Using gradient descent with learning rate η, the logit update is:

$$\Delta l_t^i(y) = \eta\hat{A}_t^i(\mathbf{1}_{y=o_t^i} - \pi_t^i(y)). \tag{16}$$

Probability Change. According the multivariate Taylor Expansion, for softmax, the first-order change in probability is:

$$\Delta\pi_t^i(y) \approx \sum_{z\in\mathcal{V}}\frac{\partial\pi_t^i(y)}{\partial l_t^i(z)}\Delta l_t^i(z) = \sum_{z\in\mathcal{V}}\pi_t^i(y)(\mathbf{1}_{z=y} - \pi_t^i(z))\Delta l_t^i(z) = \pi_t^i(y)\left(\Delta l_t^i(y) - \sum_{z\in\mathcal{V}}\pi_t^i(z)\Delta l_t^i(z)\right) \tag{17}$$

where $\pi_t^i(y) = \frac{exp(l_t^i(y))}{\sum_{z\in\mathcal{V}} exp(l_t^i(z))}$. Substituting $\Delta\pi_t^i(y)$.

$$\begin{aligned}\Delta\pi_t^i(y) &= \pi_t^i(y)\left(\eta\hat{A}_t^i(\mathbf{1}_{y=o^i} - \pi_t^i(y)) - \sum_{z\in\mathcal{V}}\pi_t^i(z)\eta\hat{A}_t^i(\mathbf{1}_{z=o_t^i} - \pi_t^i(z))\right) \\ &= \eta\hat{A}_t^i\pi_t^i(y)\left(\mathbf{1}_{y=o_t^i} - \pi_t^i(y) - \pi_t^i(o_t^i) + \sum_{z\in\mathcal{V}}\pi_t^i(z)^2\right).\end{aligned} \tag{18}$$

Entropy Change. According the Taylor Expansion, the first-order change in entropy H_t^i is:

$$\Delta H_t^i \approx \sum_{y\in\mathcal{V}}\frac{\partial H_i^t}{\partial\pi_t^i(y)}\Delta\pi_t^i(y) = -\sum_y\Delta\pi_t^i(y)(1 + \log\pi_t^i(y)), \quad \text{where} \quad H_i^t = \left(-\sum_{y\in\mathcal{V}}\pi_t^i(y)\log\pi_t^i(y)\right). \tag{19}$$

Substituting $\Delta\pi_t^i(y)$:

$$\begin{aligned}\Delta H_t^i &= -\eta\hat{A}_t^i \sum_{y\in\mathcal{V}} \pi_t^i(y)\left[\mathbf{1}_{y=o_t^i} - \pi_t^i(y) - \pi_t^i(o_t^i) + \sum_{z\in\mathcal{V}} \pi_t^i(z)^2\right](1+\log\pi_t^i(y)) \\ &= -\eta\hat{A}_t^i\left[\pi_t^i(o_t^i)\log\pi_t^i(o_t^i) - \sum_{y\in\mathcal{V}} \pi_t^i(y)^2\log\pi_t^i(y) - \pi_t^i(o_t^i)\sum_{y\in\mathcal{V}} \pi_t^i(y)\log\pi_t^i(y) + \sum_{z\in\mathcal{V}} \pi_t^i(z)^2\sum_{y\in\mathcal{V}} \pi_t^i(y)\log\pi_t^i(y)\right] \\ &= -\eta\hat{A}_t^i\left[\pi_t^i(o_t^i)\log\pi_t^i(o_t^i) - \sum_{y\in\mathcal{V}} \pi_t^i(y)^2 H_t^i - \sum_{y\in\mathcal{V}} \pi_t^i(y)^2\log\pi_t^i(y) + \pi_t^i(o_t^i)H_t^i\right]\end{aligned} \tag{20}$$

Average Entropy Change. *Ignore the impact of cross-context updates*, the first-order change in token entropy over all queries is:

$$\Delta H_{avg} = \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o^i|}\sum_{t=1}^{|o^i|} -\eta\hat{A}_t^i\left(\pi_t^i(o_t^i)\log\pi_t^i(o_t^i) - \sum_{y\in\mathcal{V}}\pi_t^i(y)^2\log\pi_t^i(y) + \pi_t^i(o_t^i)H_t^i - \sum_{y\in\mathcal{V}}\pi_t^i(y)^2 H_t^i\right)\right] \tag{21}$$

Let $B_t^i = \pi_t^i(o_t^i)\log\pi_t^i(o_t^i) + \pi_t^i(o_t^i)H_t^i$ and $F_t^i = -\sum_{y\in\mathcal{V}}\pi_t^i(y)^2\log\pi_t^i(y) - \sum_{y\in\mathcal{V}}\pi_t^i(y)^2 H_t^i$:

$$\Delta H_{avg} = \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{-\eta}{G}\sum_{i=1}^{G}\mathbb{E}_{t=1:|o^i|}\left[\hat{A}_t^i\left(B_t^i + F_t^i\right)\right]\right] \tag{22}$$

The response with maximum PPL of $\{o^i\}_{i=1}^G$ is denoted as o^m,

- Rewarding maximum PPL $\hat{A}_t^m = \sqrt{G-1}$, $\hat{A}_t^i = -\frac{1}{\sqrt{G-1}}$.

$$\begin{aligned}\Delta H_{avg} &= \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{-\eta}{G}\left(-\frac{1}{\sqrt{G-1}}\sum_{\substack{i=1\\ i\neq m}}^{G}\mathbb{E}_{t=1:|o^i|}\left[\left(B_t^i + F_t^i\right)\right] + \sqrt{G-1}\mathbb{E}_{t=1:|o^m|}\left[\left(B_t^m + F_t^m\right)\right]\right)\right] \\ &= \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{-\eta\sqrt{G-1}}{G}\left(\mathbb{E}_{t=1:|o^m|}\left[B_t^m + F_t^m\right] - \mathbb{E}_{i\neq m}\left[\mathbb{E}_{t=1:|o^i|}\left[B_t^i + F_t^i\right]\right]\right)\right] \\ &= \frac{-\eta\sqrt{G-1}}{G}\left(\mathbb{E}\left[B_t^m\right] + \mathbb{E}\left[F_t^m\right] - \mathbb{E}_{i\neq m}\left[B_t^i\right] - \mathbb{E}_{i\neq m}\left[F_t^i\right]\right)\end{aligned} \tag{23}$$

  We assume the difference in F_t is negligible as it reflects global distribution statistics that remain statistically invariant across samples in a large dataset, thus $\mathbb{E}\left[F_t^m\right] - \mathbb{E}_{i\neq m}\left[F_t^i\right] \approx 0$.

$$\Delta H_{avg} \approx \frac{\eta\sqrt{G-1}}{G}\left(\mathbb{E}_{i\neq m}\left[B_t^i\right] - \mathbb{E}\left[B_t^m\right]\right) \tag{24}$$

- Penalizing maximum PPL $\hat{A}_t^m = -\sqrt{G-1}$, $\hat{A}_t^i = \frac{1}{\sqrt{G-1}}$.

$$\Delta H_{avg} \approx \frac{\eta\sqrt{G-1}}{G}\left(\mathbb{E}\left[B_t^m\right] - \mathbb{E}_{i\neq m}\left[B_t^i\right]\right) \tag{25}$$

When $e^{-1-H_t^i} < \pi_t^i(o_t^i) < 1$, B_t^i increases with $\pi_t^i(o_t^i)$. Moreover, since $\{o^i\}$ is sampled by the π with probability, and the smaller H_t^i is, the higher the sampling probability, we can approximately assume that $e^{-1-H_t^i} < \pi_t^i(o_t^i) < 1$ is satisfied. **These approximations suggest that rewarding the maximum-PPL response tends to increase the average token entropy, whereas penalizing it tends to decrease the average token entropy.**. □

Import Statement. Given that the parameter space of LLMs is vast and complex updating, strict mathematical proof is unrealizable. The aforementioned proof is based on multiple idealized assumptions and only provide an approximate estimate of the trend in entropy change. In actual training, the theoretical results need to be further verified through experiments.

A.2 Experimental verification

To further verify the feasibility of the theory in section 3.3, we designed a verification experiment. Specifically, we used the Qwen3-0.6B Yang et al. [2025a] model and employed DAPO-17K Yu et al. [2025] as the training set, discarding the verification reward. Instead, we utilize max-PPL reward and max-PPL penalty as training reward, respectively, and recorded the changes in model entropy. As illustrated in Figure 5, the trend of entropy update is consistent with the proof.

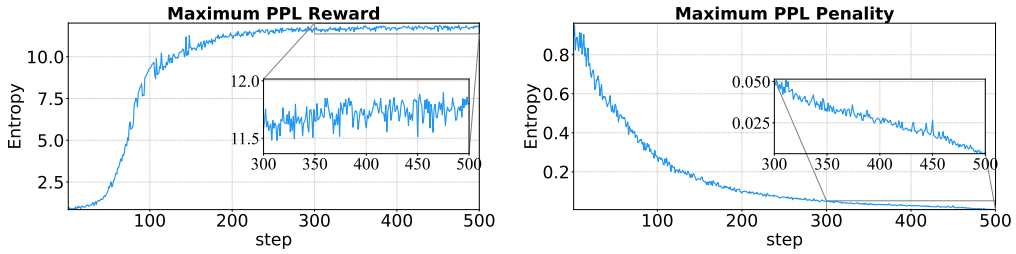


Figure 5: Entropy curves of maximum-PPL reward and maximum-PPL penalty trained on Qwen3-0.6B model with using DAPO-17K.

B Related Works

B.1 Reinforcement Learning for LLMs

Policy optimization algorithms have evolved significantly to address the challenges of RL for LLMs (LLM-RL). Trust Region Policy Optimization (TRPO) Schulman et al. [2015] introduced KL divergence constraints for stable policy updates, laying foundational ideas. Proximal Policy Optimization (PPO) Schulman et al. [2017] simplified TRPO with clip-based objectives, becoming the standard for LLM-RL. GRPO Shao et al. [2024] optimized PPO for mathematical reasoning by a novel advantage calculation strategy, boosting efficiency and removing value networks. DAPO Li et al. [2025] further scaled LLM-RL with decoupled clipping and dynamic sampling, while VAPO Yue et al. [2025] enhanced reasoning reliability via hierarchical advantage estimation. GSPO Zheng et al. [2025] extended GRPO's grouping strategy to sequence-level optimization for mixture-of-experts models and long-form reasoning tasks.

B.2 Reward Shaping

Recent works on reward shaping for LLM-RL have advanced across key directions. CrossDomain-RLVR Su et al. [2025] expanded verifiable reward RL to diverse unstructured domains via generative scoring. PKPO Walder and Karkhanis [2025] and Pass@k Training Chen et al. [2025b] optimized pass@k performance to enhance sample diversity and collective utility, addressing exploration limitations. rl-without-gt Xin et al. [2025] introduced format-length surrogate signals to bypass ground truth dependence. RuscaRL Zhou et al. [2025] leveraged rubric scaffolding to break exploration bottlenecks, while DACE Li et al. [2025], CDE Dai et al. [2025] proposed difficulty-aware certainty and curiosity-driven signals for adaptive exploration. DRER He et al. [2025] focused on reasoning quality with fine-grained CoT rewards, and OBE Song et al. [2025] mitigated diversity collapse via outcome-based exploration bonuses.

B.3 Entropy and Perplexity-Driven Exploration

Entropy and perplexity are key signals for RL of adaptive balance between exploration and exploitation. Some advantage-enhanced methods Cheng et al. [2025], Deng et al. [2025] revisited

entropy/perplexity as a signal, augmenting the advantage function to promote deep reasoning chains and boost Pass@K. Liu et al. [2025] extended entropy mechanisms to test-time RL via ETMR and EAR, enhancing efficiency and diversity. Dai et al. [2025] integrated actor perplexity and critic value variance as curiosity bonuses to mitigate premature convergence. Li et al. [2025] leveraged difficulty-aware certainty to dynamically modulate exploration, rewarding or penalizing confidence based on task complexity. Chen et al. [2025a] introduced adaptive unlearning in two-stage rollouts to break entropy collapse loops.

C Algorithm Details

Algorithm 1 Perplexity Space Disentangling

Require: Batch of queries $\{\boldsymbol{q}\}$ from $\mathcal{D}$, Policy $\pi_{\boldsymbol{\theta}}$, PPL Queue $\mathcal{Q}$

for each query $\boldsymbol{q}$ **do**
  // G-Sampling
  Generate outputs $\{\boldsymbol{o}^i\}_{i=1}^{G} \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{q})$
  Compute verifiable rewards $\{\boldsymbol{r}^i\} \leftarrow \mathcal{R}(\boldsymbol{o}^i, \boldsymbol{a})$
  Compute perplexity $\{\boldsymbol{p}^i\}$ using Eq. (4)
  Update $\mathcal{Q}$ with pairs $\{(\boldsymbol{p}^i, \boldsymbol{r}^i)\}$
end for
Estimate probabilities $\widehat{\Pr}(R|P > \tau)$ and $\widehat{\Pr}(R|P < \tau)$ using Eq. (5)
Calculate candidate set $\mathcal{C} = \{\tau | \Delta_{\mathrm{EiS}}(\tau) > 0 \wedge \Delta_{\mathrm{ErS}}(\tau) > 0\}$ using (Eq. 7 Eq. 8)
if $\mathcal{C} \neq \emptyset$ **then**
  $\tau^* \leftarrow \arg\min_{\tau \in \mathcal{C}} \frac{1}{|\mathcal{Q}|} \sum_{(\boldsymbol{r}_i, \boldsymbol{p}_i) \in \mathcal{Q}} |\boldsymbol{r}_j - \mathbb{I}(\boldsymbol{p}_j < \tau)|$
else
  $\tau^* \leftarrow$ None
end if

Algorithm 2 Bidirectional Reward Reallocation

Require: Optimal threshold $\tau*$, A group of verification reward $\{\boldsymbol{r}^i\}$ and corresponding PPL $\{\boldsymbol{p}^i\}$

Initialize reallocated rewards $\{\boldsymbol{r}_r^i\} \leftarrow \{\boldsymbol{r}^i\}$
if τ^* is not None **then**
  $p = \text{mean}(\{\boldsymbol{p}^i\})$
  $m = \arg\max_i \boldsymbol{p}^i$
  if $\forall i, \boldsymbol{r}^i = 0$ **and** $p < \tau^*$ **then**
    // Hard Group in EiS
    $\boldsymbol{r}_r^m \leftarrow 1$ {Encourage Exploration}
  else if $\forall i, \boldsymbol{r}^i = 1$ **and** $p > \tau^*$ **then**
    // Easy Group in ErS
    $\boldsymbol{r}_r^m \leftarrow 0$ {Encourage Exploitation}
  else if $\exists i, \boldsymbol{r}^i = 0$ **and** $\exists j, \boldsymbol{r}^j = 1$ **then**
    $\{\boldsymbol{r}_r\} \leftarrow \{0\}$
  end if
end if

D Detailed Experiment Setup

D.1 Mathematical Reasoning

The mathematical reasoning task uses the DAPO-17K Yu et al. [2025] dataset for 700 steps training on 4×8 A100 GPUs. For all comparison methods, all parameters are set from Table 5, and the remaining parameters use the default parameters of VERL Sheng et al. [2025]. For DAPO/w EL, use 0.001 as the coefficient for entropy loss, and the coefficient for entropy loss for the other methods is all 0. For the reproduced CDE Dai et al. [2025], their exclusive hyper-parameters all adopt the default settings from the original paper.

The prompt template for the training set and test set is as: "[Question] Let's think step by step and output the final answer within \boxed{}", [Question] represents a specific mathematical problem.

Table 5: Training configurations of mathematical reasoning.

data	train_batch_size	128	actor	ppo_mini_batch	128	rollout	temperature	1.2
	max_prompt_length	1K		clip_ratio_low	0.2		n	8
	max_response_length	4K		clip_ratio_high	0.28		val_kwargs.n	8
	gen_batch_size	256		kl_loss_coef	0		val_kwargs.temperature	0.6

D.2 Function Calling.

The baseline for Function Calling is TooRL Qian et al. [2025], which is the earliest open-source method to introduce GRPO into Function Calling. Its main contribution lies in a series of reward designs related to function calling, where the reward is no longer a binary 0 or 1 but a range of $[-3, 4]$. The validation rewards of TooRL includes formatting scores and match scores, match scores further includes tool name matching, parameter name matching, and parameter content matching. In order to adapt to DiPO, we denoted the sample with the maximum reward as the correct sample, and other samples as error samples. During BRR, we use 4 and -3 for the correct and error reward reallocation respectively. We implement DAPO and DiPO on 8 A100 GPUs, the hyperparameter settings for DAPO and DiPO are shown in Table 6, which are consistent with TooRL except DAPO's private parameters. The training dataset and system prompt are both consistent with TooRL.

Table 6: Training configurations of function calling.

data	train_batch_size	512	actor	ppo_mini_batch	128	rollout	temperature	1.0
	max_prompt_length	2K		clip_ratio_low	0.2		n	4
	max_response_length	1K		clip_ratio_high	0.28		val_kwargs.n	4
	gen_batch_size	1024		kl_loss_coef	0		val_kwargs.temperature	0.6

E More Experiment Results

E.1 Robustness Across Training Runs

To examine sensitivity to training randomness, we compare DiPO with DAPO+EL on Qwen3-4B-Base over three independent training runs. Table 7 reports both best-checkpoint and last-checkpoint results on six mathematical reasoning benchmarks, using the ACC/mean@8 metric defined for the main evaluation. We evaluate the best checkpoint and the last checkpoint for each training run. For each run, AVG is the unweighted average across the six benchmarks. We report the mean and standard error of the mean (SEM) across the included training runs. For a score x_r from run r, these are computed as

$$\mu = \frac{1}{n}\sum_{r=1}^{n} x_r, \qquad \mathrm{SEM} = \frac{s}{\sqrt{n}}, \qquad s^2 = \frac{1}{n-1}\sum_{r=1}^{n}(x_r - \mu)^2. \tag{26}$$

Under best-checkpoint evaluation, DiPO improves AVG over DAPO+EL by 0.64 percentage points and achieves higher AVG in all three runs. DiPO also exhibits lower across-run variability in AVG. These results support the reproducibility of the aggregate improvement, although the gains are uneven across individual benchmarks.

For last-checkpoint evaluation, the training log shows that the third DAPO+EL run suffered a training collapse at step 660, whereas DiPO did not collapse in Run3. We therefore retain this result in the table but exclude the entire Run3 comparison from the summary statistics, using the same two runs for both methods. On these non-collapsed paired runs, DiPO improves AVG over DAPO+EL by 0.70 percentage points. Thus, the positive average difference persists without relying on the collapsed baseline run. This conditional comparison summarizes the non-collapsed runs; the observed collapse remains a separate stability observation. Given the small number of runs, these results should not be interpreted as establishing statistical significance across training seeds.

Table 7: Training-run robustness on Qwen3-4B-Base. Within each checkpoint panel, the higher and lower method means are marked in **bold** and underlined, respectively; SEM values and individual runs are not ranked.

Method	Run / statistic	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base: best checkpoint ($n = 3$)								
DAPO+EL	Run1	26.67	24.58	86.78	62.95	54.53	44.53	50.01
	Run2	27.50	24.17	86.75	62.20	53.62	44.44	49.78
	Run3	25.83	24.17	86.30	62.80	54.23	44.30	49.61
	$\mu \pm$ SEM	26.67 $\pm$ 0.48	24.31 $\pm$ 0.14	86.61 $\pm$ 0.16	62.65 $\pm$ 0.23	**54.13** $\pm$ 0.27	44.42 $\pm$ 0.07	49.80 $\pm$ 0.12
DiPO (ours)	Run1	29.17	24.58	87.00	63.70	54.09	44.76	50.55
	Run2	27.08	24.17	86.93	64.91	53.78	44.72	50.27
	Run3	28.33	25.83	86.28	65.21	53.57	43.84	50.51
	$\mu \pm$ SEM	**28.19** $\pm$ 0.61	**24.86** $\pm$ 0.50	**86.74** $\pm$ 0.23	**64.61** $\pm$ 0.46	53.81 $\pm$ 0.15	**44.44** $\pm$ 0.30	**50.44** $\pm$ 0.09
Qwen3-4B-Base: last checkpoint ($n = 2$ for summaries)								
DAPO+EL	Run1	21.67	25.00	85.30	59.34	51.84	42.56	47.62
	Run2	17.92	20.83	86.38	59.79	52.62	43.29	46.81
	Run3[†]	6.67	3.75	49.10	24.40	25.22	23.35	22.08
	$\mu \pm$ SEM	19.80 $\pm$ 1.88	**22.92** $\pm$ 2.09	**85.84** $\pm$ 0.54	59.57 $\pm$ 0.23	52.23 $\pm$ 0.39	**42.93** $\pm$ 0.37	47.22 $\pm$ 0.41
DiPO (ours)	Run1	24.58	21.25	86.35	61.45	52.79	42.65	48.18
	Run2	24.58	20.42	85.30	60.09	52.93	42.60	47.65
	Run3[‡]	25.00	25.83	85.45	59.79	52.45	41.54	48.34
	$\mu \pm$ SEM	**24.58** $\pm$ 0.00	20.84 $\pm$ 0.42	85.83 $\pm$ 0.53	**60.77** $\pm$ 0.68	**52.86** $\pm$ 0.07	42.63 $\pm$ 0.03	**47.92** $\pm$ 0.27

Table 8: Effect of the perplexity queue size on Qwen3-4B-Base. B denotes the training batch size.

Queue size	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base							
$2B$ (default)	**29.17**	24.58	**87.00**	**63.70**	54.09	44.76	**50.55**
$10B$	27.50	**25.00**	86.78	63.25	**54.46**	**44.94**	50.32

E.2 Effect of the Perplexity Queue Size

PSD estimates its perplexity threshold from a queue of recent observations. To assess sensitivity to the amount of retained history, we compare the default queue size of $2B$ with a larger queue size of $10B$, where B denotes the training batch size. Table 8 reports the results on Qwen3-4B-Base.

Increasing the queue size from $2B$ to $10B$ reduces AVG by 0.23 percentage points. The larger queue improves AIME25, OLY, and MIN while slightly reducing MATH and AMC performance and producing a larger decrease on AIME24. The relatively small change in AVG suggests that the final aggregate performance is not highly sensitive to these two queue-size settings. This is a limited comparison rather than a comprehensive sensitivity study, so it does not establish invariance to arbitrary queue sizes. We retain $2B$ as the default setting.

E.3 PSD Activation and Threshold Dynamics

To investigate the behavior of the online threshold estimator, we record both the fraction of training steps at which PSD is activated and the estimated threshold at selected training steps. Figure 6(a) reports the activation ratios.

PSD has a lower activation ratio early in training: 48% over the first 50 steps and 74% over the first 100 steps. In each reported window from step 101 onward, the activation ratio is 100%. These results indicate that PSD becomes consistently active as training progresses, rather than being continuously active from the start.

Figure 6(b) shows the threshold trajectory. The threshold initially fluctuates, increasing from 1.109092 at step 40 to 1.181621 at step 60. It subsequently decreases to 1.093314 at step 200 and then changes more gradually, reaching 1.032716 at step 700. Together with the activation ratios, these observations suggest a more stable threshold-estimation regime in the middle and later stages of training, while still allowing the threshold to adapt to the evolving perplexity distribution.

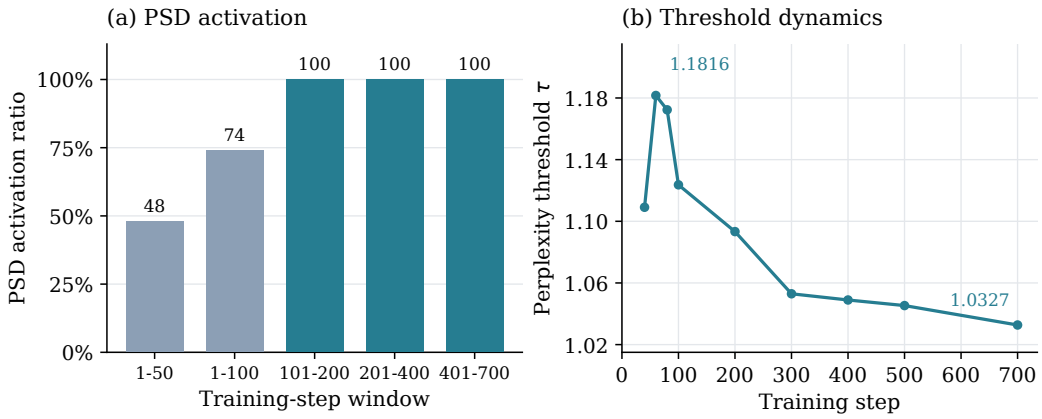


Figure 6: PSD activation and threshold dynamics. (a) PSD activation ratios over training-step windows. (b) Estimated perplexity thresholds at selected training steps. Only valid threshold estimates are shown.

E.4 Results on Llama3.1-8B-Instruct

To further verify the generality of DiPO, we conduct experiments using Llama3.1-8B-Instruct Grattafiori et al. [2024] on GSM8K Cobbe et al. [2021] and MATH, as shown in Table 9. It can be seen that for Llama3.1-8B-Instruct, entropy loss and CDE are not effective methods, their results are lower than the baseline DAPO, especially for MATH; by contrast, our method shows moderate improvement. Concretely, DiPO achieves the best MATH and AVG performance, improving over DAPO by 1.00 and 0.57 percentage points, respectively.

E.5 Results of Majority Vote

The ACC/maj@8 metric, which determines the final answer by majority voting across 8 reasoning attempts, primarily reflects a model's consistency and reliability in mathematical problem-solving. A higher ACC/maj@8 score indicates that the model can reliably generate correct reasoning paths, not just occasionally produce the right answer.

As presented in Table 10, the proposed DiPO method yields the highest average performance across the six mathematical benchmarks for all three base models investigated. For the Qwen3-4B-Base model, DiPO outperforms the second-best method, CDE, by 0.90 percentage points in AVG. It attains the highest scores, including ties, on AIME24, AIME25, AMC and MIN, while maintaining strong competitiveness on the remaining benchmarks. For the Qwen3-8B-Base model, DiPO exceeds the second-best method, CDE, by 0.96 percentage points in AVG and achieves the top performance on four benchmarks, namely AIME24, AIME25, AMC and OLY. For the Qwen2.5-7B model, DiPO outperforms the second-best method, DAPO, by 0.82 percentage points in AVG and ranks first on three benchmarks: AIME24, AMC and OLY. Collectively, DiPO attains the top performance,

Table 9: Comparison of mathematical reasoning in acc/mean@8 using Llama3.1-8B-Instruct as base model. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

Method	GSM8K	MATH	AVG
Llama3.1-8B-Instruct	79.80	47.83	63.82
DAPO	<u>89.89</u>	<u>55.75</u>	<u>72.82</u>
DAPO w/ EL	89.66	52.68	71.17
CDE	89.01	53.45	71.23
DiPO (ours)	**90.03**	**56.75**	**73.39**

Table 10: Comparison of mathematical reasoning in ACC/maj@8 on 6 mathematics benchmarks. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

Model	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base							
Base model	13.33	6.67	64.00	46.99	37.59	30.15	33.12
GRPO	**36.67**	26.67	89.60	68.67	58.10	50.37	55.01
DAPO	33.33	**30.00**	89.60	69.88	59.29	49.26	55.23
DAPO w/ EL	33.33	**30.00**	**90.40**	68.67	**60.03**	50.40	55.47
CDE	**36.67**	**30.00**	89.80	69.88	57.06	49.63	55.51
DiPO (ours)	**36.67**	**30.00**	90.20	**72.29**	58.59	**50.70**	**56.41**
Qwen3-8B-Base							
Base model	16.67	13.33	80.20	59.04	44.57	36.76	41.76
GRPO	36.67	**33.33**	92.40	77.11	62.11	**54.41**	59.34
DAPO	36.67	**33.33**	92.40	77.11	61.07	53.68	59.04
DAPO w/ EL	40.00	**33.33**	92.00	78.31	61.37	52.21	59.54
CDE	40.00	**33.33**	**93.20**	75.90	62.11	53.68	59.70
DiPO (ours)	**43.33**	**33.33**	92.60	**79.52**	**62.90**	52.25	**60.66**
Qwen2.5-7B							
Base model	6.67	3.33	52.20	27.71	25.85	20.59	22.73
GRPO	26.67	**26.67**	84.40	65.06	46.95	41.18	48.49
DAPO	30.00	23.33	83.40	**67.47**	47.70	41.91	48.97
DAPO w/ EL	30.00	20.00	83.40	63.86	48.14	**43.01**	48.07
CDE	23.33	20.00	**85.00**	63.86	49.33	41.18	47.12
DiPO (ours)	**33.33**	23.33	84.00	**67.47**	**49.44**	41.18	**49.79**

including ties, in 11 out of the 18 benchmark-model pairs, which validates its consistent and superior capability in generating reliable mathematical reasoning processes.

E.6 Coefficient Sensitivity Analysis

As reported in Table 11, DiPO is less sensitive to its coefficient than entropy loss. On Qwen3-8B-Base, DiPO improves AVG over DAPO by 1.56 percentage points at a coefficient of 0.10 and remains above the baseline at 1.00. Increasing this coefficient tenfold reduces AVG by only 1.43 points. In contrast, increasing the entropy-loss coefficient from 0.001 to 0.01 reduces AVG by 7.90 points, placing it 7.23 points below the baseline, with degradation across all benchmarks. On Qwen3-4B-Base, DiPO also maintains relatively stable performance between coefficients of 0.10 and 1.00, whereas entropy loss at 0.01 substantially degrades performance. These results indicate that DiPO offers a wider effective coefficient range than entropy loss.

E.7 Results of Risk Prediction

In the contemporary landscape of digital platform management, real-time public opinion risk prediction is a critical capability for safeguarding the operational integrity and brand reputation. This task involves a complex analysis of user-generated content to identify potential crises across various dimensions, such as product experience, fraud, and regulatory compliance. It can be realized through LLMs guided by a specialized prompt that mandates a "risk-first" priority to ensure that no negative sentiment is overlooked. As detailed in the 7, the LLM is instructed to maintain strict standards for "no-risk" classification, where any ambiguity or negative indicator automatically triggers a risk label. Due to confidentiality reasons, the content within the brackets "[]" in the prompt is replaced with placeholders.

To evaluate our approach in risk prediction, we constructed 4000 data (3000 for training and 1000 for test), and conducted a comparative analysis of DiPO and DAPO, trained on Qwen3-8B model Yang et al. [2025a]. As reported in Table 12, DiPO improves accuracy, recall, and F1 over DAPO

Table 11: The impact of the coefficients for DiPO and entropy loss on the results. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

Mathod	coeff	AIME24	AIME25	MATH	AMC	OLY	MIN	AVG
Qwen3-4B-Base								
Baseline	0.00	26.25	23.75	86.43	61.90	53.88	44.34	49.43
DiPO	0.10	**29.17**	**24.58**	**87.00**	**63.70**	54.09	44.76	**50.55**
	1.00	25.83	24.17	86.38	62.05	54.01	**45.50**	49.66
DAPO /w EL	0.001	26.67	**24.58**	86.78	62.95	**54.53**	44.53	50.01
	0.01	18.33	20.00	81.12	50.00	45.99	39.38	42.47
Qwen3-8B-Base								
Baseline	0.00	30.08	25.83	89.43	69.12	56.90	48.02	53.23
DiPO	0.10	**35.00**	**27.50**	89.55	**71.23**	**57.73**	47.75	**54.79**
	1.00	32.08	24.58	88.68	70.03	56.71	**48.07**	53.36
DAPO /w EL	0.001	33.75	25.42	**89.58**	69.87	57.21	47.56	53.90
	0.01	22.50	20.42	84.18	56.33	49.49	43.06	46.00

by 1.43, 1.91, and 1.04 percentage points, respectively, with a 0.27-point decrease in precision. The improvement in recall is particularly relevant to risk prediction, where missing critical risks is costly.

You are a professional Public Opinion Emergency Analyst for **[The Company]**. You need to analyze given public opinion information related to **[The Company]**. Based on your role and the standards provided below for risk/no-risk labels, determine whether the public opinion contains a risk, and provide the specific category and the basis for your judgment.

Notes:
- Prioritize identifying risk labels. The criteria for "No Risk" categories must be strictly met and the information must be non-negative. If the content meets both risk and no-risk criteria, or is in an ambiguous state, classify it as a "Risk" category.
- Ensure the strict accuracy of "No Risk" judgments.
- Ensure the output format is correct for subsequent data processing and auditing.
- Please conduct your reasoning step-by-step. Wrap your final answer in <answer></answer>, where the final answer is risk for risk or norisk for no risk.

Public Opinion Content: **[user-generated content]**

Risk Categories and Criteria:
- **[Risk name]**: **[Risk description]**.
- ...
- Other Risk: Other negative public opinion involving **[The Company]**.

No-Risk Categories and Criteria:
- **[No Risk name]**: **[No risk description]**.
- ...
- Others: Positive public opinion related to **[The Company]** that does not involve regulatory department.

Figure 7: Prompt of risk prediction.

Table 12: The results of risk prediction, including accuracy, precision, recall, and F1 score, and all results are the mean of 8 independent inferences. Best and second-best distinct scores within each model block are marked in **bold** and underlined, respectively; ties share the same marking.

Method	ACC/mean@8	Precision/mean@8	Recall/mean@8	F1/mean@8
Qwen3-8B	52.06	87.18	54.64	67.18
DAPO	76.94	**95.96**	77.58	85.80
DiPO	**78.37**	95.69	**79.49**	**86.84**

E.8 Visualization of PPL Distribution

To further analyze the exploration and exploitation trends during the RL training process, we conducted a visualization analysis of the PPL distribution during the training of DAPO and DiPO. As shown in Figure 8, during the initial training stage, the PPL distributions of DAPO and DiPO are relatively consistent, and it is difficult to distinguish the PPL distributions of correct and error samples, which is the **motivation for introducing advantage judgment** in PSD. In the later stages of training, the PPL distributions of DAPO and DiPO gradually converge. The difference is that for DAPO, the overall PPL will converge to a lower range (whether it is error samples or correct samples), while PPL

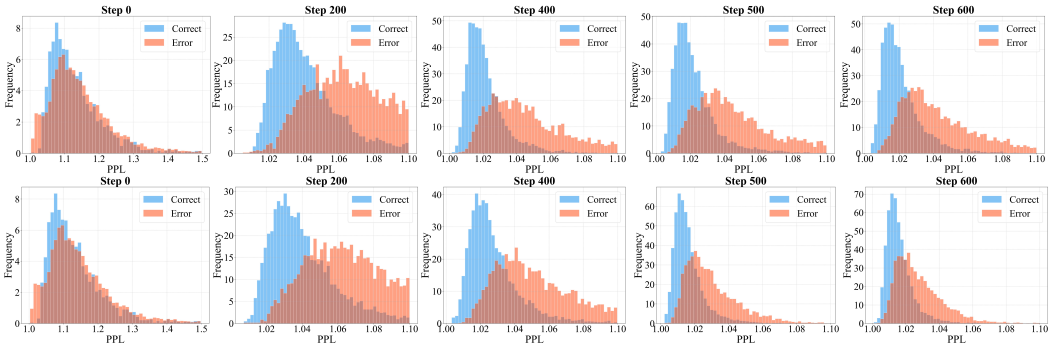


Figure 8: PPL distribution of correct and error samples for Qwen3-8B-Base trained on DAPO-17K Dataset via DiPO (TOP) and DAPO (Bottom).

Query: Roslyn has ten boxes. Five of the boxes contain pencils, four of the boxes contain pens, and two of the boxes contain both pens and pencils. How many boxes contain neither pens nor pencils? Let's think step by step and output the final answer within boxed{}.

Alright, I have this problem in front of me:\n\n**Roslyn has ten boxes. Five of the boxes contain pencils, four of the boxes contain pens, and two of the boxes contain both pens and pencils. How many boxes contain neither pens nor pencils?**\n\nAt first glance, it seems like a typical set theory problem. I'll try to break it down step by step.\n\n### Understanding the Problem\n\nFirst, let's summarize the information:\n\n- **Total number of boxes:** 10\n- **Boxes that contain pencils:** 5\n- **Boxes that contain pens:** 4\n- **Boxes that contain both pens and pencils:** 2\n\nWe need to find out how many boxes contain **neither** pens **nor** pencils.\n\n### Visualizing the Problem\n\nI think using a Venn diagram might help here. Let's draw two circles that overlap:\n\n- One circle represents boxes with **pencils**.\n- The other circle represents boxes with **pens**.\n- The overlapping area represents boxes that have both pens and pencils.\n\nGiven that:\n\n- The number of boxes with **both** pens and pencils is 2.\n\nSo, the overlapping area is 2.\n\n### Finding Boxes with Only Pencils\n\nTotal boxes with pencils = boxes with only pencils + boxes with both.\n\nGiven that total boxes with pencils = 5, and boxes with both = 2.\n\nSo, boxes with **only** pencils = 5 - 2 = **3**.\n\n### Finding Boxes with Only Pens\n\nSimilarly, total boxes with pens = boxes with only pens + boxes with both.\n\nGiven that total boxes with pens = 4, and boxes with both = 2.\n\nSo, boxes with **only** pens = 4 - 2 = **2**.\n\n### Calculating Boxes with Pens or Pencils or Both\n\nNow, the boxes that have **either** pens **or** pencils **or** both would be:\n\nBoxes with only pencils + boxes with only pens + boxes with both.\n\nThat is, 3 (only pencils) + 2 (only pens) + 2 (both) = **7** boxes.\n\n### Finding Boxes with Neither Pens nor Pencils\n\nTotal boxes = 10.\n\nBoxes with pens or pencils or both = 7.\n\nTherefore, boxes with **neither** = Total boxes - boxes with pens or pencils or both.\n\nSo, 10 - 7 = **3**.\n\n### Double-Checking the Work\n\nLet's make sure everything adds up:\n\n- Only pencils: 3\n- Only pens: 2\n- Both: 2\n- Neither: 3\n\nAdding them up: 3 + 2 + 2 + 3 = 10. That matches the total number of boxes. So, it seems correct.\n\n### Final Answer\n\nAfter working through the problem, the number of boxes that contain neither pens nor pencils is:\n\n\[\boxed{3}\]<|endoftext|>

Figure 9: Correct case of DAPO.

Query: Roslyn has ten boxes. Five of the boxes contain pencils, four of the boxes contain pens, and two of the boxes contain both pens and pencils. How many boxes contain neither pens nor pencils? Let's think step by step and output the final answer within boxed{}.

Alright, I have this problem about Roslyn having ten boxes. Some contain pencils, some contain pens, and a few contain both. I need to find out how many boxes contain neither pens nor pencils. Let's break it down.\n\n**Given:**\n- Total number of boxes = 10\n- Boxes that contain pencils = 5\n- Boxes that contain pens = 4\n- Boxes that contain both pens and pencils = 2\n\n**Find:**\n- Number of boxes that contain neither pens nor pencils.\n\nAlright, I need to find out how many boxes have neither pens nor pencils. To do this, I can use the principle of inclusion-exclusion.\n\nFirst, let's find out how many boxes contain either pens or pencils or both.\n\nThe formula for the number of boxes that contain pencils or pens or both is:\n\n\[\text{Number with pencils or pens or both} = \text{Number with pencils} + \text{Number with pens} - \text{Number with both} \]\n\nPlugging in the numbers:\n\n\[\text{Number with pencils or pens or both} = 5 + 4 - 2 = 7 \]\n\nSo, 7 boxes contain either pens or pencils or both.\n\nSince there are 10 boxes in total, the number of boxes that contain neither pens nor pencils is:\n\n\[\text{Total boxes} - \text{Boxes with pencils or pens or both} = 10 - 7 = 3 \]\n\nTherefore, the number of boxes that contain neither pens nor pencils is \boxed{3}\]<|endoftext|>

Figure 10: Correct case of DiPO.

distribution of DiPO is more discriminative, with error samples in a higher PPL range and correct samples in a lower PPL range, reflecting a reasonable tendency for exploration and exploitation.

E.9 Case Analysis

Figure 10 9 12 11 show some cases of extreme groups for DAPO and DiPO trained to the 500-th step on DAPO-17K, where darker colors indicate higher entropy. It can be seen that under correct answers, the overall entropy of DiPO cases is smaller, and under incorrect answers, the density of high-entropy tokens in DiPO is greater, which also reflects the effective exploitation and exploration balance of DiPO.

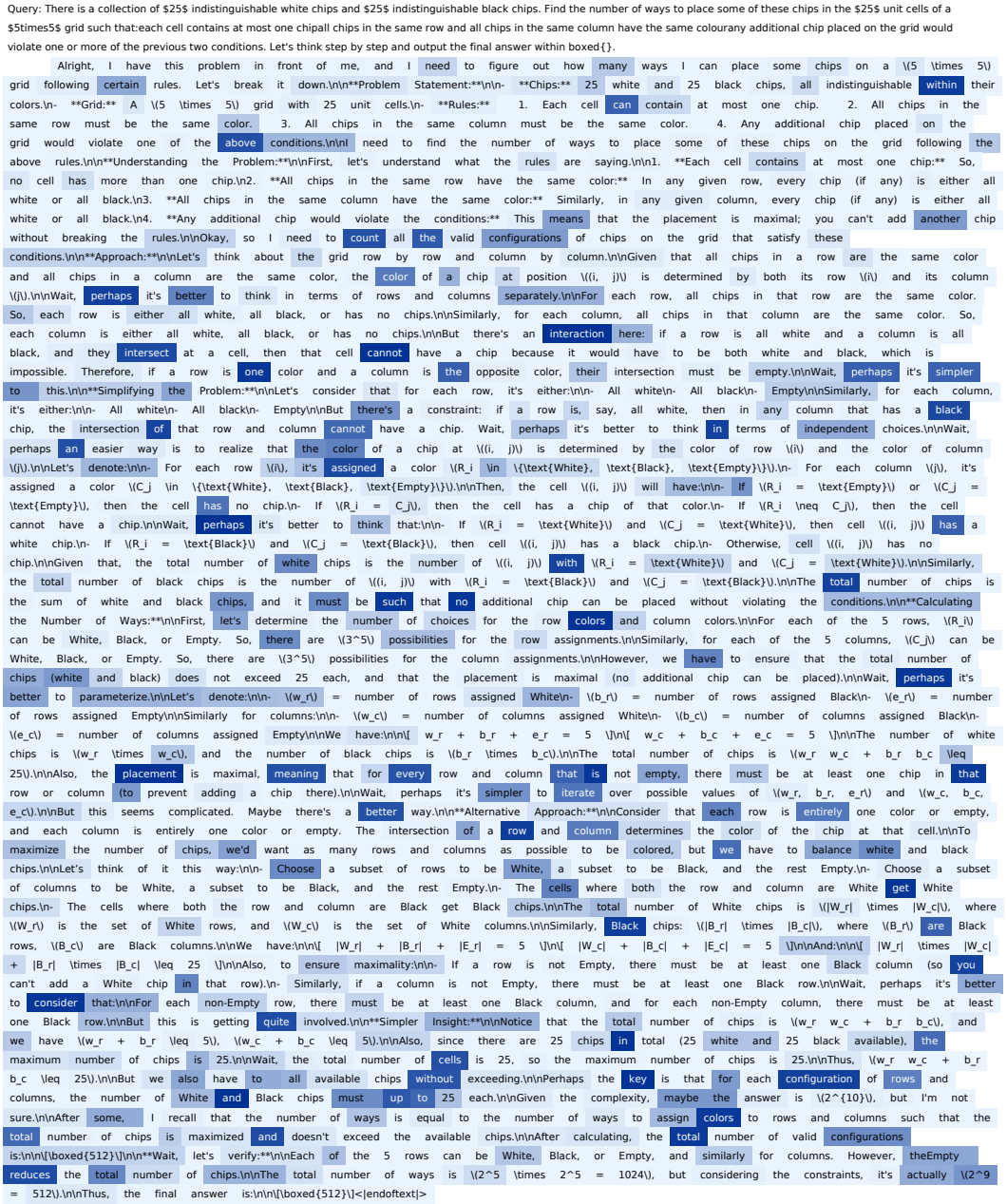

Query: There is a collection of 25 indistinguishable white chips and 25 indistinguishable black chips. Find the number of ways to place some of these chips in the 25 unit cells of a 5×5 grid such that:each cell contains at most one chipall chips in the same row and all chips in the same column have the same colourany additional chip placed on the grid would violate one or more of the previous two conditions. Let's think step by step and output the final answer within boxed{}.
Alright, I have this problem in front of me, and I need to figure out how many ways I can place some chips on a \(5 \times 5\) grid following certain rules. Let's break it down.\n\n**Problem Statement:**\n\n- **Chips:** 25 white and 25 black chips, all indistinguishable within their colors.\n- **Grid:** A \(5 \times 5\) grid with 25 unit cells.\n- **Rules:** 1. Each cell can contain at most one chip. 2. All chips in the same row must be the same color. 3. All chips in the same column must be the same color. 4. Any additional chip placed on the grid would violate one of the above conditions.\n\nI need to find the number of ways to place some of these chips on the grid following the above rules.\n\n**Understanding the Problem:**\n\nFirst, let's understand what the rules are saying.\n\n1. **Each cell contains at most one chip:** So, no cell has more than one chip.\n2. **All chips in the same row have the same color:** In any given row, every chip (if any) is either all white or all black.\n3. **All chips in the same column have the same color:** Similarly, in any given column, every chip (if any) is either all white or all black.\n4. **Any additional chip would violate the conditions:** This means that the placement is maximal; you can't add another chip without breaking the rules.\n\nOkay, so I need to count all the valid configurations of chips on the grid that satisfy these conditions.\n\n**Approach:**\n\nLet's think about the grid row by row and column by column.\n\nGiven that all chips in a row are the same color and all chips in a column are the same color, the color of a chip at position \((i, j)\) is determined by both its row \(i\) and its column \(j\).\n\nWait, perhaps it's better to think in terms of rows and columns separately.\n\nFor each row, all chips in that row are the same color. So, each row is either all white, all black, or has no chips.\n\nSimilarly, for each column, all chips in that column are the same color. So, each column is either all white, all black, or has no chips.\n\nBut there's an interaction here: if a row is all white and a column is all black, and they intersect at a cell, then that cell cannot have a chip because it would have to be both white and black, which is impossible. Therefore, if a row is one color and a column is the opposite color, their intersection must be empty.\n\nWait, perhaps it's simpler to this.\n\n**Simplifying the Problem:**\n\nLet's consider that for each row, it's either:\n\n- All white\n- All black\n- Empty\n\nBut there's a constraint: if a row is, say, all white, then in any column that has a black chip, the intersection of that row and column cannot have a chip. Wait, perhaps it's better to think in terms of independent choices.\n\nWait, perhaps an easier way is to realize that the color of a chip at \((i, j)\) is determined by the color of row \(i\) and the color of column \(j\).\n\nLet's denote:\n\n- For each row \(i\), it's assigned a color \(R_i \in \{\text{White}, \text{Black}, \text{Empty}\}\).\n- For each column \(j\), it's assigned a color \(C_j \in \{\text{White}, \text{Black}, \text{Empty}\}\).\n\nThen, the cell \((i, j)\) will have:\n\n- If \(R_i = \text{Empty}\) or \(C_j = \text{Empty}\), then the cell has no chip.\n- If \(R_i = C_j\), then the cell has a chip of that color.\n- If \(R_i \neq C_j\), then the cell cannot have a chip.\n\nWait, perhaps it's better to think that:\n\n- If \(R_i = \text{White}\) and \(C_j = \text{White}\), then cell \((i, j)\) has a white chip.\n- If \(R_i = \text{Black}\) and \(C_j = \text{Black}\), then cell \((i, j)\) has a black chip.\n- Otherwise, cell \((i, j)\) has no chip.\n\nGiven that, the total number of white chips is the number of \((i, j)\) with \(R_i = \text{White}\) and \(C_j = \text{White}\).\n\nSimilarly, the total number of black chips is the number of \((i, j)\) with \(R_i = \text{Black}\) and \(C_j = \text{Black}\).\n\nThe total number of chips is the sum of white and black chips, and it must be such that no additional chip can be placed without violating the conditions.\n\n**Calculating the Number of Ways:**\n\nFirst, let's determine the number of choices for the row colors and column colors.\n\nFor each of the 5 rows, \(R_i\) can be White, Black, or Empty. So, there are \(3^5\) possibilities for the row assignments.\n\nSimilarly, for each of the 5 columns, \(C_j\) can be White, Black, or Empty. So, there are \(3^5\) possibilities for the column assignments.\n\nHowever, we have to ensure that the total number of chips (white and black) does not exceed 25 each, and that the placement is maximal (no additional chip can be placed).\n\nWait, perhaps it's better to parameterize.\n\nLet's denote:\n\n- \(w_r\) = number of rows assigned White\n- \(b_r\) = number of rows assigned Black\n- \(e_r\) = number of rows assigned Empty\n\nSimilarly for columns:\n\n- \(w_c\) = number of columns assigned White\n- \(b_c\) = number of columns assigned Black\n- \(e_c\) = number of columns assigned Empty\n\nWe have:\n\n\[w_r + b_r + e_r = 5 \]\n\[w_c + b_c + e_c = 5 \]\n\nThe number of white chips is \(w_r \times w_c\), and the number of black chips is \(b_r \times b_c\).\n\nThe total number of chips is \(w_r w_c + b_r b_c \leq 25\).\n\nAlso, the placement is maximal, meaning that for every row and column that is not empty, there must be at least one chip in that row or column (to prevent adding a chip there).\n\nWait, perhaps it's simpler to iterate over possible values of \(w_r, b_r, e_r\) and \(w_c, b_c, e_c\).\n\nBut this seems complicated. Maybe there's a better way.\n\n**Alternative Approach:**\n\nConsider that each row is entirely one color or empty, and each column is entirely one color or empty. The intersection of a row and column determines the color of the chip at that cell.\n\nTo maximize the number of chips, we'd want as many rows and columns as possible to be colored, but we have to balance white and black chips.\n\nLet's think of it this way:\n\n- Choose a subset of rows to be White, a subset to be Black, and the rest Empty.\n- The cells where both the row and column are White get White chips.\n- The cells where both the row and column are Black get Black chips.\n\nThe total number of White chips is \(|W_r| \times |W_c|\), where \(W_r\) is the set of White rows, and \(W_c\) is the set of White columns.\n\nSimilarly, Black chips: \(|B_r| \times |B_c|\), where \(B_r\) are Black rows, \(B_c\) are Black columns.\n\nWe have:\n\n\[|W_r| + |B_r| + |E_r| = 5 \]\n\[|W_c| + |B_c| + |E_c| = 5 \]\n\nAnd:\n\n\[|W_r| \times |W_c| + |B_r| \times |B_c| \leq 25 \]\n\nAlso, to ensure maximality:\n\n- If a row is not Empty, there must be at least one Black column (so you can't add a White chip in that row).\n- Similarly, if a column is not Empty, there must be at least one Black row.\n\nWait, perhaps it's better to consider that:\n\nFor each non-Empty row, there must be at least one Black column, and for each non-Empty column, there must be at least one Black row.\n\nBut this is getting quite involved.\n\n**Simpler Insight:**\n\nNotice that the total number of chips is \(w_r w_c + b_r b_c\), and we have \(w_r + b_r \leq 5\), \(w_c + b_c \leq 5\).\n\nAlso, since there are 25 chips in total (25 white and 25 black available), the maximum number of chips is 25.\n\nWait, the total number of cells is 25, so the maximum number of chips is 25.\n\nThus, \(w_r w_c + b_r b_c \leq 25\).\n\nBut we also have to all available chips without exceeding.\n\nPerhaps the key is that for each configuration of rows and columns, the number of White and Black chips must up to 25 each.\n\nGiven the complexity, maybe the answer is \(2^{10}\), but I'm not sure.\n\nAfter some, I recall that the number of ways is equal to the number of ways to assign colors to rows and columns such that the total number of chips is maximized and doesn't exceed the available chips.\n\nAfter calculating, the total number of valid configurations is:\n\n\[\boxed{512}\]\n\n**Wait, let's verify:**\n\nEach of the 5 rows can be White, Black, or Empty, and similarly for columns. However, theEmpty reduces the total number of chips.\n\nThe total number of ways is \(2^5 \times 2^5 = 1024\), but considering the constraints, it's actually \(2^9 = 512\).\n\nThus, the final answer is:\n\n\[\boxed{512}\]<|endoftext|>">

Figure 11: Error case of DAPO.

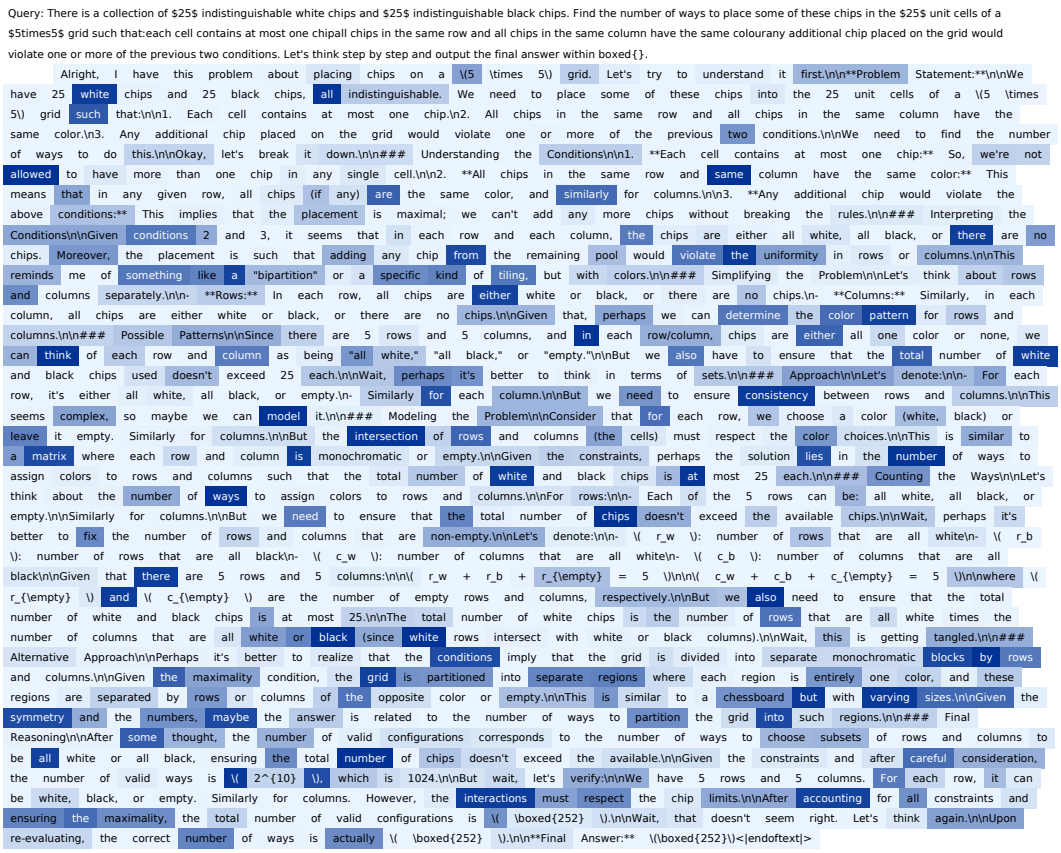


Figure 12: Error case of DiPO.