

Don't Peek at the Answer: Outcome-Masked Group Relative Policy Optimization for Label-Free RLVR

Yongshi Ye^{1,3*}, Liang Zhang^{2*}, Yidong Chen^{2,3}, Xiaodong Shi^{1,2,3†}, Biao Fu^{2,3†}

¹Institute of Artificial Intelligence, Xiamen University

²School of Informatics, Xiamen University

³Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism
{yeyongshi, biaofu}@stu.xmu.edu.cn, mandel@xmu.edu.cn

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) improves LLM reasoning but typically relies on ground-truth (GT) answers, limiting scalability. Voting-based label-free RLVR replaces gold supervision with answer-level consensus from model samples. However, collapse arises when the same answer-level signal is used both to estimate rewards and to drive token-level policy optimization, encouraging the model to directly reinforce answer tokens rather than improve reasoning. We propose OM-GRPO, a label-free RLVR framework that decouples reward estimation from policy optimization. OM-GRPO masks gradients on the answer span while retaining answer-level rewards through a soft consensus signal, shifting optimization pressure away from answer tokens. We further introduce Contrast-Augmented Reward, which refines reward estimation via low-cost pairwise comparisons over existing trajectories without additional rollouts. Across diverse reasoning benchmarks and three LLM backbones, OM-GRPO outperforms label-free RLVR methods, matches supervised GT-Reward training with stable optimization, and gains 4.24 points over majority voting in Test-Time Training.

1 Introduction

As large language models (LLMs) evolve from pattern matching toward complex reasoning (Jaeche et al., 2024; DeepSeek-AI et al., 2025; Yang et al., 2025b), reinforcement learning with verifiable rewards (RLVR) has emerged as a central paradigm in post-training. RLVR improves reasoning by optimizing outcome-based rewards reflecting final-answer correctness, which are typically computed using curated ground-truth answers (Shao et al., 2024; Yu et al., 2025a). Despite these promising

advances, this reliance on high-quality human annotations makes reward construction costly and difficult to scale with training data (Ouyang et al., 2022; Shao et al., 2024; Yue et al., 2025).

Recent studies explore label-free approaches that derive rewards directly from model outputs (Prabhudesai et al., 2025; Zhao et al., 2025a). A representative strategy is majority-voting-based self-rewarding (Shafayat et al., 2025; Zhang et al., 2025): the model samples multiple rollouts for the same prompt, extracts their final answers, and rewards trajectories whose answers agree with the group consensus. However, this design introduces a fundamental issue. The answer-level consensus is used not only to estimate reward, but also to drive token-level policy optimization. Consequently, the model can improve reward by directly sharpening high-frequency answer tokens, without improving the reasoning that leads to them. This shortcut makes voting-based label-free RLVR particularly vulnerable to reward hacking and mode collapse.

To empirically examine this issue, we evaluate this class of self-rewarding methods across multiple LLM backbones. As shown in Figure 1, these methods often exhibit early performance improvements followed by a sharp degradation as training progresses. To further diagnose this behavior, we track the diversity of extracted final answers throughout training. The right panel of Figure 1 shows that the performance collapse coincides with a rapid reduction in answer diversity, with the model eventually converging to the same final answer across entire batches and even across different batches. Overall, these results indicate that the observed collapse arises from over-optimization of answer tokens.

In this work, we propose Outcome-Masked Group Relative Policy Optimization (OM-GRPO), a simple and effective framework that improves reasoning by optimizing reasoning trajectories quality rather than answer-level agreement. Specifically, OM-GRPO uses a soft reward (based on answer

* Equal contribution.

† Corresponding authors.

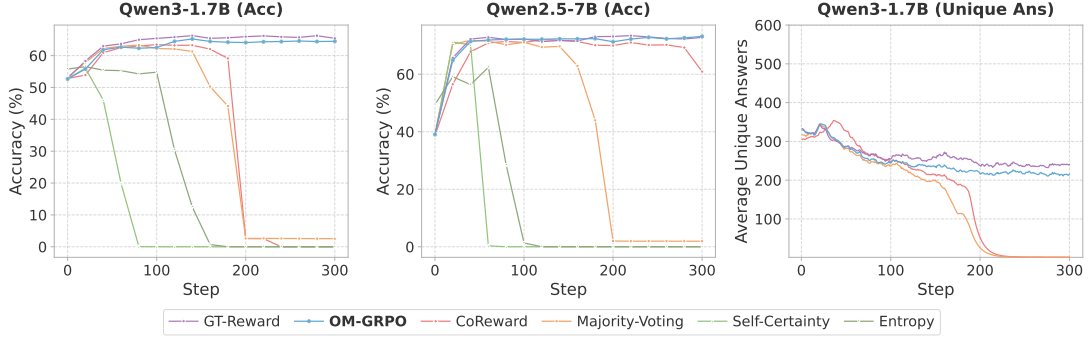


Figure 1: Training dynamics of different methods on the MATH5000 validation set. Left and middle: validation accuracy over training steps. Right: average number of unique final answers per batch during training.

frequency within the group) and masks the answer span during gradient updates, ensuring that learning signals are confined to the reasoning trajectory. Under this design, reward improvements cannot be achieved by directly sharpening or repeating answer tokens, but must arise from improved reasoning. As a result, OM-GRPO exhibits stable, non-collapsing training dynamics across backbones (see Figure 1 and 3). Notably, OM-GRPO also achieves performance comparable to GT-Reward across backbones (see Figure 1). We attribute this gain to the soft reward, which provides an implicit contrastive signal by assigning higher credit to answers that receive broader support within the group.

To amplify contrastive gains, we further introduce Contrast-Augmented Reward (CAR) to better capture the relative quality of reasoning trajectories. CAR explicitly constructs trajectory-level pairwise comparisons within each group by prompting the model to compare pairs of reasoning traces from the same input and produce only a short final answer to the original question. This yields a large number of additional outcomes, enabling more reliable estimation of soft rewards for the original trajectories without re-sampling long reasoning trajectories. Intuitively, consider two trajectories (y_1, z_1) and (y_2, z_2) for the same input, where y_1 represents higher-quality reasoning. Through pairwise comparison, if the model favors y_1 and generates z_1 , then z_1 appears more frequently within the group, thereby strengthening the soft reward assigned to y_1 . By expanding the answer pool via low-cost pairwise comparisons, this comparison-based signal yields more reliable reward estimates, resulting in more stable optimization toward higher-quality reasoning trajectories without regenerating full reasoning processes. Across diverse reasoning benchmarks and three LLM backbones, OM-GRPO con-

sistently outperforms prior label-free RLVR baselines while remaining competitive with supervised GT-Reward. This stability extends to Test-Time Training, where OM-GRPO prevents collapse and outperforms majority voting by 4.24 points.

Contributions. (1) We identify an overlooked failure mode in label-free RLVR, where answer-token shortcuts cause reward hacking and training collapse. (2) We propose OM-GRPO, a label-free RLVR framework that mitigates reward hacking via answer-span gradient masking. (3) We further introduce CAR, a low-cost strategy for refining reward estimation through pairwise comparisons. (4) Across multiple backbones and benchmarks, OM-GRPO achieves stable training and competitive or superior performance.

2 Related Work

LLM Reasoning. LLMs have achieved remarkable progress across diverse tasks (Grattafiori et al., 2024; Yang et al., 2025b), yet their reasoning outputs often remain fluent but logically unreliable (Ouyang et al., 2022; Rafailov et al., 2023; Touvron et al., 2023). Prior work improves reasoning through prompting strategies such as chain-of-thought (CoT) and self-consistency (Wei et al., 2022; Wang et al., 2023), which encourage intermediate reasoning and aggregate multiple sampled solutions. More recently, RLVR has emerged as an effective reasoning post-training paradigm (Shao et al., 2024; DeepSeek-AI et al., 2025; Lambert et al., 2024; Hu et al., 2025; Kimi-Team et al., 2025; Yang et al., 2025a; Chen et al., 2026b), using GT solutions or programmatic verifiers to reward answer correctness. Despite strong gains, its reliance on curated supervision limits scalability.

Label-Free RLVR. To scale RLVR beyond curated supervision, label-free methods derive training sig-

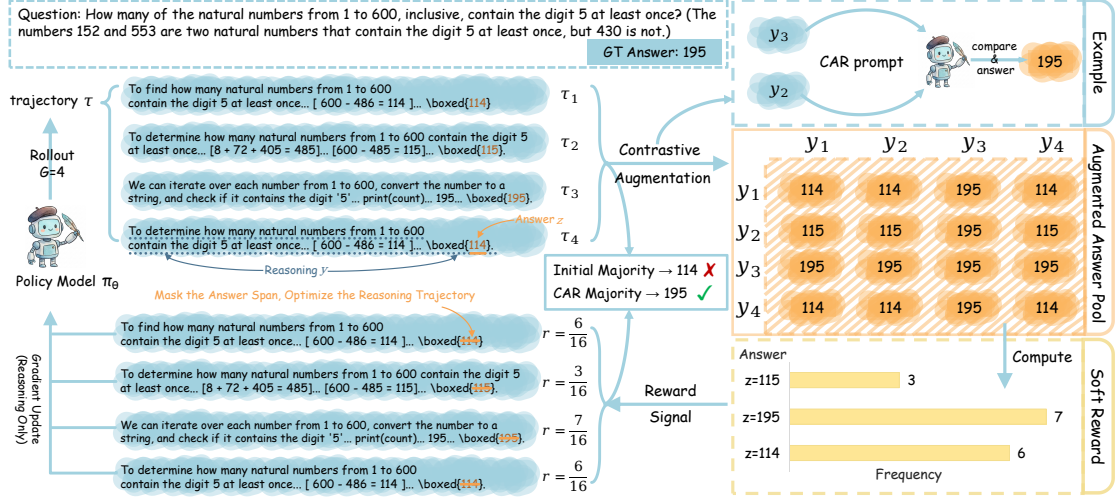


Figure 2: Overview of OM-GRPO with outcome masking updates and contrast-augmented reward estimation.

nals without gold answers (Chen et al., 2026a). Existing approaches use answer agreement (Shafayat et al., 2025; Zhang et al., 2025), uncertainty-based signals (Prabhudesai et al., 2025; Agarwal et al., 2025; Zhao et al., 2025b), or external LLM and symbolic feedback (Pang et al., 2023; Lee et al., 2024; Su et al., 2025; Zhao et al., 2025a). Among these, voting-based self-rewarding assigns rewards by group consensus (Shafayat et al., 2025; Zhang et al., 2025). However, using the same answer-level signal for reward estimation and token-level optimization creates a shortcut: the policy can reinforce answer tokens rather than reasoning, leading to reward hacking and mode collapse. Although regularization (Yu et al., 2025b) and multi-view distillation (Zhang et al., 2025) improve stability, they do not eliminate this shortcut.

3 Method: Outcome-Masked GRPO

We propose OM-GRPO, a label-free RLVR algorithm that optimizes reasoning trajectories without answer-level shortcuts. For each input x , OM-GRPO samples a group of G trajectories $\{\tau_i\}_{i=1}^G$ and updates the policy using a group-relative objective with masked gradients:

$$\mathcal{J}_{\text{OM-GRPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\tau_i|} m_{i,t} \left(\min(\rho_{i,t}(\theta) \hat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i) - \beta \text{KL}_{i,t}(\pi_\theta \| \pi_{\text{ref}}) \right) \right] \quad (1)$$

where $\rho_{i,t} = \frac{\pi_\theta(o_{i,t}|o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|o_{i,<t})}$ is the importance sampling ratio. The advantage term $\hat{A}_i$ is computed by standardizing the rewards within each group to reduce variance: $\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r}) + \delta}$. Crucially, $m_{i,t}$ is a binary mask set to 0 for tokens in the answer span z_i and 1 otherwise. We next introduce the key components of OM-GRPO.

3.1 Reward Formulation

Soft Reward. Given an input question x , the policy π_θ samples a trajectory $\tau = (y, z)$, where y denotes the chain-of-thought (CoT) reasoning and z denotes the span of answer tokens inside $\boxed{\cdot}$, excluding the formatting tokens. In the absence of gold labels, a common approach is to evaluate correctness via MV (Wang et al., 2023), which assigns a hard reward based on whether an answer matches the majority outcome within a group of sampled trajectories. However, this binary signal is sparse and unstable, motivating a soft reward based on group-level answer frequency. Given a group of G sampled trajectories, we define the soft reward for trajectory τ_i as the probability mass assigned to its answer z_i within the group of sampled answers:

$$r_i^q = \Pr(z = z_i | z \in (z_1, \dots, z_G)) \quad (2)$$

Unlike hard MV, this frequency-based signal retains relative comparisons among candidate answers instead of enforcing a binary consensus.

Format Reward. To ensure the model produces parseable outputs, we incorporate a format reward r_i^f . This is a binary indicator that equals 1 if the response contains a valid extractable $\boxed{\cdot}$ span, and 0 otherwise.

Final Reward. The final reward for τ_i combines the soft and format rewards: $r_i = r_i^q + r_i^f$.

3.2 Outcome-Masked Update

As discussed in Section 1, existing label-free RLVR methods suffer from a critical failure mode: the policy tends to exploit a trivial shortcut by directly increasing the probability of high-frequency answer tokens, regardless of the input. This leads to mode collapse, where the model memorizes answers rather than learning to reason¹. To address this, we introduce the Outcome-Masked Update (OMU), which blocks this shortcut by preventing gradient backpropagation through the answer span z . As a result, gradient updates are applied only to the reasoning chain y and the format tokens, encouraging the policy to increase the likelihood of correct answers through better reasoning².

Formally, given an input problem x , the policy π_θ samples a group of G trajectories $\{\tau_i\}_{i=1}^G$ from the old policy $\pi_{\theta_{old}}$. Each trajectory $\tau_i = (y_i, z_i)$ consists of a reasoning chain y_i and a final answer span z_i . We define a binary mask $\mathbf{m}_i$ with $m_{i,t} = 0$ on answer-span tokens z_i and $m_{i,t} = 1$ otherwise. Intuitively, although rewards are computed from answer-level statistics, masking prevents direct optimization of answer tokens and forces learning to improve the reasoning trajectory instead.

3.3 Contrast-Augmented Reward

While increasing the number of sampled trajectories G improves the reliability of voting-based rewards, it incurs prohibitive training cost for long CoT reasoning (Wang et al., 2023). We therefore propose Contrast-Augmented Reward (CAR), an efficient alternative that strengthens the relative signal of soft rewards through pairwise comparison without additional rollouts.

Augmented Answer Pool. We first construct an augmented answer pool $\mathcal{Z}_{aug}$ via pairwise comparison. For each pair of trajectories (τ_i, τ_j) within the group, we construct a contrastive prompt (Figure 8) that presents the two candidate reasoning traces and asks the model to produce a final answer for the original question. Crucially, the model generates only a short answer $\hat{z}_{ij}$, rather than re-generating a full reasoning trajectory, resulting in negligible additional cost. By enumerating all pairwise combinations with $i \neq j$, this procedure expands the

augmented answer pool from G to $O(G^2)$ while reusing existing trajectories. Formally, the augmented answer pool is defined as:

$$\mathcal{Z}_{aug} = \{z_i\}_{i=1}^G \cup \{\hat{z}_{ij}\}_{i \neq j} \quad (3)$$

Soft Reward Estimation. We then use the augmented answer pool $\mathcal{Z}_{aug}$ to estimate the soft reward of each original trajectory τ_i . Specifically, the soft reward r_i^q is defined as the probability mass assigned to its answer z_i within the augmented pool:

$$r_i^q = \Pr(z = z_i \mid z \in \mathcal{Z}_{aug}). \quad (4)$$

Importantly, the pairwise-generated answers $\hat{z}_{ij}$ serve only as validators for reward estimation and are not treated as additional training trajectories in policy-gradient updates.

This contrastive mechanism yields two key benefits. First, answers arising from fragile reasoning or lucky guessing are unlikely to be consistently reproduced across pairwise comparisons, causing their proportion in $\mathcal{Z}_{aug}$ to be diluted. Second, through pairwise comparison, robust reasoning traces are repeatedly favored over weaker alternatives and assigned higher rewards, which in turn reinforce the learning of better reasoning trajectories.

4 Experiments

4.1 Setting

Datasets and Evaluation Metrics. We train on 7,500 problems from the MATH training split (Hendrycks et al., 2021) and validate on the MATH validation set (MATH5000). We evaluate on a diverse suite of benchmarks spanning mathematical reasoning, code generation, and general capabilities, including AIME, MATH, GSM8K, AMC, LiveCodeBench, CRUX, IFEval, and MMLU-Pro. Evaluation protocols and metric definitions are provided in Appendix A.

Backbone Models. Our experiments are conducted on a set of open-source backbone language models, including Qwen2.5-7B (Yang et al., 2025b), Qwen3-1.7B-Base (Yang et al., 2025a), and Llama-3.2-3B-Instruct (Grattafiori et al., 2024). These backbones cover multiple architectures and model scales, enabling us to evaluate the robustness of our method across different backbones. Implementation details are provided in Appendix B.

¹Theoretical proof provided in Appendix D.1.

²See theoretical proof in Appendix D.2.

Methods	Mathematics					Code		Instruction	Multi-Task	Average
	AIME24	AIME25	MATH-500	GSM8K	AMC	LiveCode	CRUX	IFEval	MMLU-Pro	
Qwen3-1.7B-Base										
Before RL	3.12 $\pm$ 1.8	1.25 $\pm$ 1.1	46.9 $\pm$ 2.9	64.20 $\pm$ 1.8	20.78 $\pm$ 2.5	4.59 $\pm$ 0.7	7.52 $\pm$ 1.1	33.66 $\pm$ 1.5	33.60 $\pm$ 0.7	23.96
GT-Reward	6.88 $\pm$ 2.1	5.21 $\pm$ 2.0	68.3 $\pm$ 2.3	82.01 $\pm$ 1.1	30.87 $\pm$ 3.3	14.84 $\pm$ 0.5	33.73 $\pm$ 0.6	38.70 $\pm$ 1.7	39.43 $\pm$ 0.7	35.55
Self-Certainty	3.54 $\pm$ 1.6	3.12 $\pm$ 1.8	57.0 $\pm$ 2.7	73.71 $\pm$ 1.4	27.41 $\pm$ 2.3	10.37 $\pm$ 0.9	18.73 $\pm$ 1.3	38.58 $\pm$ 1.7	35.28 $\pm$ 0.7	29.75
Entropy	6.67 $\pm$ 1.6	4.17 $\pm$ 1.5	65.0 $\pm$ 1.7	79.45 $\pm$ 1.6	31.17 $\pm$ 3.0	12.64 $\pm$ 1.0	30.38 $\pm$ 0.5	35.00 $\pm$ 1.6	37.01 $\pm$ 0.7	33.50
Majority-Voting	2.08 $\pm$ 1.3	2.08 $\pm$ 1.3	59.6 $\pm$ 0.9	81.08 $\pm$ 0.8	29.82 $\pm$ 1.7	14.29 $\pm$ 0.5	32.00 $\pm$ 2.3	37.16 $\pm$ 1.6	35.68 $\pm$ 0.7	32.64
CoReward	6.88 $\pm$ 1.5	3.12 $\pm$ 1.5	65.4 $\pm$ 1.2	81.07 $\pm$ 0.8	33.13 $\pm$ 3.0	14.27 $\pm$ 0.5	34.10 $\pm$ 1.1	37.28 $\pm$ 1.6	37.39 $\pm$ 0.7	34.74
OM-GRPO	5.62 $\pm$ 1.8	2.50 $\pm$ 1.4	67.3 $\pm$ 0.8	82.09 $\pm$ 0.6	34.49 $\pm$ 3.8	14.60 $\pm$ 0.8	34.50 $\pm$ 1.4	39.53 $\pm$ 1.7	36.08 $\pm$ 0.7	35.19
Llama-3.2-3B-Instruct										
Before RL	6.88 $\pm$ 2.1	0.42 $\pm$ 0.6	44.4 $\pm$ 2.1	68.78 $\pm$ 1.2	17.47 $\pm$ 3.3	3.00 $\pm$ 0.4	24.70 $\pm$ 1.2	54.41 $\pm$ 1.8	32.01 $\pm$ 0.7	28.01
GT-Reward	10.62 $\pm$ 2.1	0.21 $\pm$ 0.4	47.5 $\pm$ 1.4	78.60 $\pm$ 0.8	22.89 $\pm$ 1.9	7.05 $\pm$ 0.6	32.48 $\pm$ 0.8	50.16 $\pm$ 1.8	34.26 $\pm$ 0.7	31.53
Self-Certainty	2.71 $\pm$ 1.3	0.62 $\pm$ 0.7	42.4 $\pm$ 0.9	73.52 $\pm$ 1.2	17.62 $\pm$ 1.8	5.57 $\pm$ 0.9	24.55 $\pm$ 1.2	54.11 $\pm$ 1.8	34.42 $\pm$ 0.7	28.39
Entropy	3.54 $\pm$ 1.5	0.21 $\pm$ 0.4	40.5 $\pm$ 2.9	68.57 $\pm$ 1.7	17.17 $\pm$ 2.6	5.38 $\pm$ 0.8	26.30 $\pm$ 1.7	54.24 $\pm$ 1.8	33.54 $\pm$ 0.7	27.72
Majority-Voting	9.79 $\pm$ 2.2	0.83 $\pm$ 0.8	46.4 $\pm$ 2.9	79.68 $\pm$ 0.4	18.83 $\pm$ 2.8	5.54 $\pm$ 0.4	30.95 $\pm$ 1.3	49.32 $\pm$ 1.7	34.32 $\pm$ 0.7	30.63
CoReward	7.71 $\pm$ 1.4	0.00 $\pm$ 0.0	47.2 $\pm$ 1.9	80.00 $\pm$ 1.7	19.58 $\pm$ 3.6	5.57 $\pm$ 0.4	30.87 $\pm$ 1.2	50.46 $\pm$ 1.8	32.95 $\pm$ 0.7	30.48
OM-GRPO	8.96 $\pm$ 1.7	0.21 $\pm$ 0.4	50.0 $\pm$ 1.8	78.58 $\pm$ 1.0	23.34 $\pm$ 2.1	4.47 $\pm$ 0.5	32.30 $\pm$ 1.1	49.01 $\pm$ 1.8	33.99 $\pm$ 0.7	31.21
Qwen2.5-7B										
Before RL	4.38 $\pm$ 1.4	1.67 $\pm$ 1.4	49.2 $\pm$ 2.6	71.00 $\pm$ 1.9	22.44 $\pm$ 3.7	4.57 $\pm$ 0.5	27.38 $\pm$ 2.3	40.61 $\pm$ 1.7	44.03 $\pm$ 0.7	29.48
GT-Reward	18.33 $\pm$ 2.3	11.25 $\pm$ 1.7	75.1 $\pm$ 1.4	90.83 $\pm$ 0.5	46.54 $\pm$ 2.5	12.78 $\pm$ 1.6	53.67 $\pm$ 1.0	41.50 $\pm$ 1.7	45.09 $\pm$ 0.7	43.90
Self-Certainty	9.79 $\pm$ 2.4	8.96 $\pm$ 1.8	72.5 $\pm$ 2.8	88.21 $\pm$ 0.5	40.51 $\pm$ 2.1	11.87 $\pm$ 0.9	53.18 $\pm$ 1.4	39.66 $\pm$ 1.6	43.89 $\pm$ 0.8	40.95
Entropy	11.04 $\pm$ 2.3	8.33 $\pm$ 2.2	73.2 $\pm$ 1.7	87.85 $\pm$ 0.6	40.51 $\pm$ 3.1	15.68 $\pm$ 0.9	51.08 $\pm$ 1.3	40.25 $\pm$ 1.7	42.61 $\pm$ 0.7	41.17
Majority-Voting	11.25 $\pm$ 2.1	4.17 $\pm$ 1.2	71.0 $\pm$ 0.7	90.52 $\pm$ 0.7	38.70 $\pm$ 1.3	18.37 $\pm$ 0.9	52.20 $\pm$ 1.1	42.72 $\pm$ 1.7	43.83 $\pm$ 0.7	41.42
CoReward	11.25 $\pm$ 1.8	3.96 $\pm$ 1.9	73.8 $\pm$ 1.1	90.16 $\pm$ 0.9	40.81 $\pm$ 1.8	10.37 $\pm$ 0.5	55.08 $\pm$ 2.4	43.75 $\pm$ 1.7	42.08 $\pm$ 0.7	41.25
OM-GRPO	14.17 $\pm$ 2.3	8.54 $\pm$ 1.9	75.0 $\pm$ 1.9	90.54 $\pm$ 0.7	46.23 $\pm$ 1.9	14.90 $\pm$ 0.8	54.65 $\pm$ 1.8	42.06 $\pm$ 1.7	43.17 $\pm$ 0.8	43.25

Table 1: RL results (%) across diverse benchmarks; darker colors denote better results within each model group.

Baselines. We compare OM-GRPO against both supervised and label-free RLVR baselines. The supervised oracle uses ground-truth rewards (**GT-Reward**). Label-free baselines construct self-rewards from model outputs, including **Majority Voting (MV)** (Shafayat et al., 2025), confidence-based rewards (**Self-Certainty**) (Zhao et al., 2025b), entropy minimization (**Entropy**) (Prabhudesai et al., 2025), and a stronger variant **CoReward** (Zhang et al., 2025) that aggregates consensus across paraphrased input variants. For baselines that collapse late, we report the best validation checkpoint; see Appendix C for details.

4.2 Main Results

Overall Performance. We report the main results in Table 1 under the avg@k setting³ across three backbones and a diverse suite of reasoning benchmarks. Overall, OM-GRPO achieves the best average performance among all label-free methods across the three backbones, while remaining competitive with GT-Reward, indicating that OM-GRPO provides a stronger learning signal than answer-centric self-rewarding alone.

³Results are reported as mean $\pm$ 95% confidence intervals.

Robustness in Mathematical Reasoning. As shown in Table 1, OM-GRPO achieves consistent improvements on both the in-distribution MATH500 benchmark and the out-of-distribution GSM8K and AMC datasets, with particularly pronounced gains on the more challenging AMC benchmark. This joint improvement suggests that the method is not merely relying on math-specific solution templates, which typically transfer poorly across differing math problem styles. Instead, the results align with the core design of OM-GRPO, which explicitly restricts optimization to the reasoning portion of each trajectory. By masking the answer span from gradient updates, the policy cannot increase reward by directly manipulating answer tokens, and is therefore pushed to refine the intermediate reasoning that supports the final answer. Appendices E.1 and E.2 report low-contamination and Pass@k results, respectively.

Generalization Beyond Math. OM-GRPO also transfers well beyond mathematics. Across non-math benchmarks, it remains among the top-performing label-free methods on all three backbones and in several cases matches or exceeds GT-Reward. For example, under the Qwen2.5-7B backbone, OM-GRPO achieves a higher average

Methods	AIME24		AIME25		MATH500		GSM8K		AMC		Avg.
	Avg@16	Pass@16	Avg@16	Pass@16	Avg@4	Pass@4	Avg@4	Pass@4	Avg@8	Pass@8	
Qwen3-1.7B-Base											
Before TTRL	3.12 $\pm$ 1.8	16.67	1.25 $\pm$ 1.1	13.33	46.9 $\pm$ 2.9	72.0	64.20 $\pm$ 1.8	89.31	20.78 $\pm$ 2.5	56.63	38.42
Majority-Voting	2.29 $\pm$ 1.2	13.33	2.71 $\pm$ 1.5	10.00	67.1 $\pm$ 1.3	72.0	89.16 $\pm$ 0.2	91.89	28.61 $\pm$ 1.7	49.40	42.65
OM-GRPO	9.58 $\pm$ 1.6	20.00	4.58 $\pm$ 1.4	13.33	69.1 $\pm$ 1.9	78.2	87.81 $\pm$ 0.7	92.12	36.30 $\pm$ 2.0	57.83	46.89
Llama-3.2-3B-Instruct											
Before TTRL	6.88 $\pm$ 2.1	23.33	0.42 $\pm$ 0.6	6.67	44.4 $\pm$ 2.1	65.0	68.78 $\pm$ 1.2	86.88	17.47 $\pm$ 3.3	49.40	36.92
Majority-Voting	6.67 $\pm$ 0.0	6.67	0.00 $\pm$ 0.0	0.00	52.5 $\pm$ 0.5	57.8	85.80 $\pm$ 1.1	89.01	21.08 $\pm$ 1.9	31.33	35.09
OM-GRPO	8.75 $\pm$ 1.9	23.33	0.83 $\pm$ 0.8	3.33	52.5 $\pm$ 1.5	61.2	87.26 $\pm$ 0.3	91.05	21.54 $\pm$ 1.1	40.96	39.08
Qwen2.5-7B											
Before TTRL	4.38 $\pm$ 1.4	30.00	1.67 $\pm$ 1.4	20.00	49.2 $\pm$ 2.6	76.2	71.00 $\pm$ 1.9	93.78	22.44 $\pm$ 3.7	63.86	43.25
Majority-Voting	15.62 $\pm$ 1.2	20.00	6.25 $\pm$ 1.1	10.00	78.4 $\pm$ 1.4	81.2	93.69 $\pm$ 0.4	94.31	40.66 $\pm$ 2.7	55.42	49.55
OM-GRPO	12.71 $\pm$ 1.3	20.00	7.50 $\pm$ 1.6	20.00	78.3 $\pm$ 1.3	83.4	93.54 $\pm$ 0.3	94.62	42.02 $\pm$ 2.7	57.83	50.99

Table 2: Test-Time Training results (%) on math reasoning benchmarks.

Methods	Mathematics					Code		Instruction	Multi-Task	Average
	AIME24	AIME25	MATH-500	GSM8K	AMC	LiveCode	CRUX	IFEval	MMLU-Pro	
Qwen3-1.7B-Base										
GT-Reward	6.88 $\pm$ 2.1	5.21 $\pm$ 2.0	68.3 $\pm$ 2.3	82.01 $\pm$ 1.1	30.87 $\pm$ 3.3	14.84 $\pm$ 0.5	33.73 $\pm$ 0.6	38.70 $\pm$ 1.7	39.43 $\pm$ 0.7	35.55
OM-GRPO	5.62 $\pm$ 1.8	2.50 $\pm$ 1.4	67.3 $\pm$ 0.8	82.09 $\pm$ 0.6	34.49 $\pm$ 3.8	14.60 $\pm$ 0.8	34.50 $\pm$ 1.4	39.53 $\pm$ 1.7	36.08 $\pm$ 0.7	35.19
w/o Soft Reward	4.58 $\pm$ 1.8	2.92 $\pm$ 1.4	67.5 $\pm$ 2.7	83.28 $\pm$ 0.6	32.83 $\pm$ 2.3	14.65 $\pm$ 0.9	33.62 $\pm$ 1.3	33.82 $\pm$ 1.6	38.60 $\pm$ 0.7	34.64
w/o CAR	7.08 $\pm$ 1.9	1.88 $\pm$ 1.4	65.0 $\pm$ 1.8	80.88 $\pm$ 0.9	31.02 $\pm$ 1.7	14.09 $\pm$ 0.6	35.00 $\pm$ 0.8	36.65 $\pm$ 1.6	37.44 $\pm$ 0.7	34.34
w/o OMU	6.67 $\pm$ 1.9	4.17 $\pm$ 1.9	64.2 $\pm$ 1.1	81.60 $\pm$ 0.9	34.34 $\pm$ 3.4	14.14 $\pm$ 0.6	33.88 $\pm$ 2.0	33.99 $\pm$ 1.5	34.78 $\pm$ 0.7	34.20
w/o CAR and OMU	2.08 $\pm$ 1.3	2.08 $\pm$ 1.3	59.6 $\pm$ 0.9	81.08 $\pm$ 0.8	29.82 $\pm$ 1.7	14.29 $\pm$ 0.5	32.00 $\pm$ 2.3	37.16 $\pm$ 1.6	35.68 $\pm$ 0.7	32.64
Llama-3.2-3B-Instruct										
GT-Reward	10.62 $\pm$ 2.1	0.21 $\pm$ 0.4	47.5 $\pm$ 1.4	78.60 $\pm$ 0.8	22.89 $\pm$ 1.9	7.05 $\pm$ 0.6	32.48 $\pm$ 0.8	50.16 $\pm$ 1.8	34.26 $\pm$ 0.7	31.53
OM-GRPO	8.96 $\pm$ 1.7	0.21 $\pm$ 0.4	50.0 $\pm$ 1.8	78.58 $\pm$ 1.0	23.34 $\pm$ 2.1	4.47 $\pm$ 0.5	32.30 $\pm$ 1.1	49.01 $\pm$ 1.8	33.99 $\pm$ 0.7	31.21
w/o Soft Reward	8.75 $\pm$ 1.6	0.21 $\pm$ 0.4	49.6 $\pm$ 0.9	78.87 $\pm$ 0.6	21.84 $\pm$ 3.1	4.78 $\pm$ 0.6	29.90 $\pm$ 1.4	50.28 $\pm$ 1.7	35.10 $\pm$ 0.7	31.04
w/o CAR	11.46 $\pm$ 1.3	0.21 $\pm$ 0.4	48.2 $\pm$ 1.6	78.64 $\pm$ 1.9	19.73 $\pm$ 2.7	4.80 $\pm$ 0.6	28.82 $\pm$ 1.9	51.90 $\pm$ 1.8	34.05 $\pm$ 0.7	30.87
w/o OMU	8.12 $\pm$ 1.1	1.67 $\pm$ 1.1	45.1 $\pm$ 1.6	78.83 $\pm$ 1.6	20.93 $\pm$ 2.1	7.91 $\pm$ 0.9	30.32 $\pm$ 0.8	49.63 $\pm$ 1.7	33.62 $\pm$ 0.7	30.68
w/o CAR and OMU	9.79 $\pm$ 2.2	0.83 $\pm$ 0.8	46.4 $\pm$ 2.9	79.68 $\pm$ 0.4	18.83 $\pm$ 2.8	5.54 $\pm$ 0.4	30.95 $\pm$ 1.3	49.32 $\pm$ 1.7	34.32 $\pm$ 0.7	30.63
Qwen2.5-7B										
GT-Reward	18.33 $\pm$ 2.3	11.25 $\pm$ 1.7	75.1 $\pm$ 1.4	90.83 $\pm$ 0.5	46.54 $\pm$ 2.5	12.78 $\pm$ 1.6	53.67 $\pm$ 1.0	41.50 $\pm$ 1.7	45.09 $\pm$ 0.8	43.90
OM-GRPO	14.17 $\pm$ 2.3	8.54 $\pm$ 1.9	75.0 $\pm$ 1.9	90.54 $\pm$ 0.7	46.23 $\pm$ 1.9	14.90 $\pm$ 0.8	54.65 $\pm$ 1.8	42.06 $\pm$ 1.7	43.17 $\pm$ 0.8	43.25
w/o Soft Reward	12.92 $\pm$ 1.8	3.96 $\pm$ 1.7	75.7 $\pm$ 2.6	89.18 $\pm$ 0.2	44.88 $\pm$ 2.3	15.51 $\pm$ 1.6	55.38 $\pm$ 0.7	41.85 $\pm$ 1.7	45.91 $\pm$ 0.7	42.81
w/o CAR	13.96 $\pm$ 1.9	8.33 $\pm$ 1.7	76.4 $\pm$ 2.9	89.31 $\pm$ 0.6	43.37 $\pm$ 1.7	15.18 $\pm$ 1.2	54.45 $\pm$ 1.1	39.44 $\pm$ 1.7	41.18 $\pm$ 0.7	42.40
w/o OMU	10.62 $\pm$ 1.7	5.42 $\pm$ 2.3	73.8 $\pm$ 2.2	90.52 $\pm$ 1.8	43.22 $\pm$ 2.7	15.87 $\pm$ 0.4	52.00 $\pm$ 1.7	43.53 $\pm$ 1.7	44.34 $\pm$ 0.7	42.15
w/o CAR and OMU	11.25 $\pm$ 2.1	4.17 $\pm$ 1.2	71.0 $\pm$ 0.7	90.52 $\pm$ 0.7	38.70 $\pm$ 1.3	18.37 $\pm$ 0.9	52.20 $\pm$ 1.1	42.72 $\pm$ 1.7	43.83 $\pm$ 0.7	41.42

Table 3: Ablation results for the proposed OM-GRPO on reasoning benchmarks.

score than GT-Reward on non-math tasks (38.70 vs. 38.26). We also observe particularly stable gains on code-related benchmarks: on both Qwen3-1.7B-Base and Llama-3.2-3B, OM-GRPO attains stronger CRUX performance than other label-free baselines. Overall, these results show that OM-GRPO generalizes well across diverse reasoning domains without access to ground-truth rewards.

4.3 Test-Time Training Performance

We evaluate OM-GRPO in a Test-Time Training (TTRL) setting using both Avg@k and Pass@k metrics. As shown in Table 2, OM-GRPO consistently outperforms the MV baseline on both

metrics across backbones. Notably, on Llama-3.2-3B-Instruct, MV leads to a decrease in Avg@k (from 36.92 to 35.09), indicating that the policy converges toward incorrect consensus. In contrast, OM-GRPO prevents this collapse, achieving robust gains in both average response quality and success rates (avg score improves to 39.08). These results suggest that masking answer tokens provides a stable learning signal for online adaptation, effectively avoiding the risks of reward hacking.

4.4 Ablation Study

Component Ablations. Table 3 presents ablation studies on the key components of OM-GRPO.

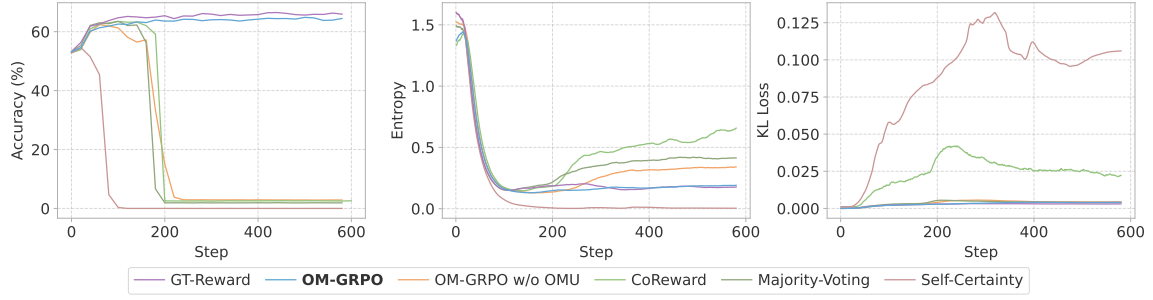


Figure 3: Training dynamics on the MATH5000 validation set, with the KL loss scaled by the KL coefficient.

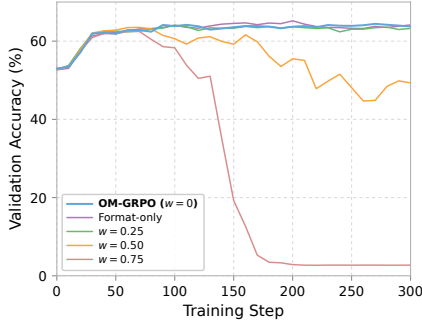


Figure 4: MATH5000 validation accuracy under answer-token update variants.

Across all backbones, the full OM-GRPO configuration consistently achieves the best average performance. Removing OMU leads to the most significant degradation across models, with average drops of 0.99 on Qwen3-1.7B-Base and 1.10 on Qwen2.5-7B, and reintroduces training collapse (see Figure 3). Furthermore, comparing w/o CAR against the vanilla MV baseline shows that OMU alone provides substantial gains (e.g., +1.70 on Qwen3-1.7B-Base). This indicates that excluding the answer span from gradient updates is critical for preventing the policy from exploiting answer-token shortcuts rather than learning the reasoning process. Similarly, replacing the soft proportional reward with a hard majority signal reduces performance, suggesting that soft rewards provide a smoother and more informative optimization signal. Overall, these results show that OM-GRPO’s gains arise from the synergy between outcome masking and robust reward calibration, enabling stable and effective learning without ground-truth labels.

Soft Answer-Token Downweighting. We further examine whether full answer-span masking can be replaced by softer answer-token gradient weighting. Figure 4 compares OM-GRPO ($w=0.0$) with variants where answer tokens receive 0.25, 0.50, or 0.75 of the outcome-reward

gradient, as well as a format-only variant in which answer tokens receive only the format reward. Small outcome-reward weights can remain stable: $w=0.25$ closely tracks OM-GRPO, and the format-only variant achieves comparable stability. However, increasing this weight makes training unstable again: $w=0.50$ gradually degrades and $w=0.75$ rapidly collapses. Thus, the risk lies in directly applying answer-level pseudo rewards to answer tokens; full masking provides a conservative way to block this feedback loop.

4.5 Why OMU Prevents Collapse

Training Stability and Collapse Prevention.

Figure 1 shows that OM-GRPO remains stable throughout training, avoiding the collapse behaviors observed in prior label-free baselines. To further stress-test this stability, we extend training to 10 epochs in Figure 3. Under prolonged optimization, different baselines exhibit distinct degeneration patterns: Self-Certainty collapses into over-confident repetition, as evidenced by spiked KL divergence and vanishing entropy; CoReward drifts toward high-entropy uncertainty; and MV undergoes consensus collapse, degenerating into trivial, high-agreement responses with near-zero accuracy and low KL divergence. In contrast, OM-GRPO exhibits remarkable stability, maintaining a performance trajectory close to the ground-truth oracle. Importantly, removing the outcome mask (w/o OMU) immediately reproduces the consensus collapse observed under MV, confirming that masking answer tokens from gradient updates is essential for preventing shortcut exploitation.

Further Analysis of Training Collapse. To further characterize this failure mode, we provide four analyses in the appendix: (1) answer-diversity dynamics to quantify mode concentration (Appendix E.3); (2) rollout-level statistics to distinguish policy-induced pseudo-label bias from in-

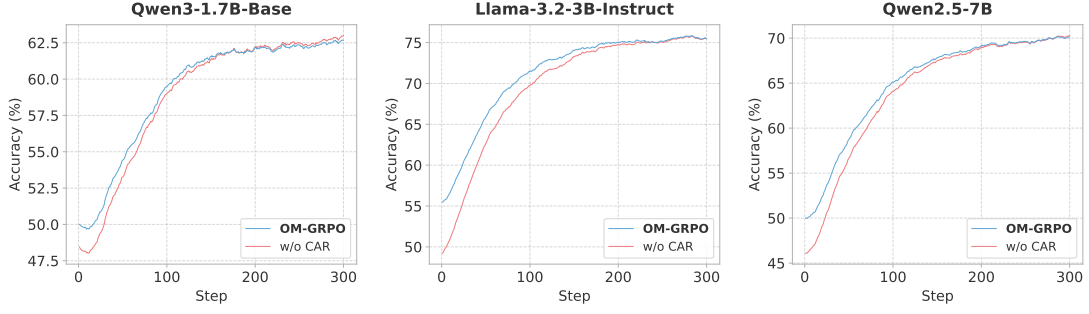


Figure 5: Impact of CAR on MATH5000 validation accuracy over training steps.

dependent noise (Appendix E.4); (3) PRM-based baselines to test whether an auxiliary process reward model prevents collapse (Appendix E.5); and (4) Qwen3-14B experiments to assess whether stronger initialization eliminates or merely delays MV degeneration (Appendix E.8).

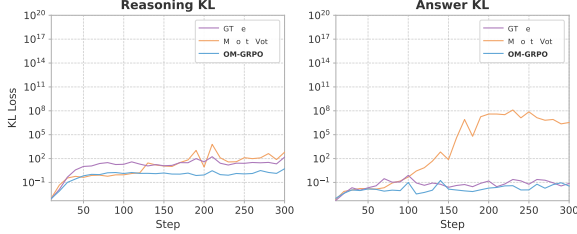


Figure 6: KL divergence of answer tokens and reasoning tokens during training for different methods.

Answer Tokens Dominate Collapse. To understand why masking only the answer span is sufficient, we analyze the training dynamics of answer and reasoning tokens separately in Figure 6. Under the MV baseline, the KL divergence on answer tokens rises sharply during training, while the KL divergence on reasoning tokens grows much more slowly. This asymmetry shows that the collapse is concentrated on the answer span rather than arising from uniform overfitting of the full trajectory. The reason is that, in voting-based label-free RLVR, rewards are assigned at the answer level, making answer tokens the most direct shortcut for increasing reward. Although they occupy only a small fraction of the output sequence, they dominate the collapse dynamics. OM-GRPO blocks this shortcut by masking gradients on the answer span. As shown in Figure 6, answer-token KL then remains low and stable, while reasoning-token KL is also well controlled. As a further check, the within-group reasoning-process consistency remains zero throughout training, indicating that collapse is not shifted from the answer span to the reasoning pro-

cess. Together, these results show that masking the answer span is sufficient to prevent collapse and preserve meaningful reasoning updates.

4.6 How CAR Improves Reward Estimation

More Reliable Voting-Based Rewards. We next evaluate whether CAR improves the reliability of voting-based reward estimation. To this end, we track the accuracy of the top-voted answer during training, measuring how often the consensus selected by voting matches the ground-truth answer. As shown in Figure 5, incorporating CAR consistently improves the accuracy of the voting-selected consensus answer across all three backbones. The gains are most pronounced in the early and middle stages of training, where vote-based rewards are particularly susceptible to noise under limited rollouts. This indicates that pairwise augmentation effectively strengthens the empirical consensus signal: by introducing solution-level comparisons through low-cost pairwise completions, CAR expands the answer pool, mitigates spurious majorities, and increases the probability that the top-voted answer matches the ground-truth answer.

Method	MATH	Others	Overall	Time/step (s)	Total Time
OM-GRPO (n=8)	38.4 $\pm$ 1.68	31.18 $\pm$ 1.15	35.19	74.74	6:13:41
w/o CAR (n=8)	37.2 $\pm$ 1.54	30.80 $\pm$ 0.93	34.34	54.43	4:32:10
w/o CAR (n=16)	37.9 $\pm$ 1.91	30.64 $\pm$ 1.14	34.66	76.88	6:24:23
w/o CAR (n=64)	38.2 $\pm$ 1.52	29.65 $\pm$ 1.33	34.42	272.23	22:41:10

Table 4: CAR vs. direct sampling across rollout budgets.

CAR vs. Scaling Rollouts. We compare CAR with directly increasing the rollout budget in Table 4. OM-GRPO with CAR (n=8) achieves the best overall average performance at a cost comparable to the compute-matched n=16 baseline: 74.74s vs. 76.88s per step and 6:13:41 vs. 6:24:23 over 300 steps. At n=64, which matches CAR’s effective answer-pool size, direct sampling costs 272.23s per step and 22:41:10 overall, making

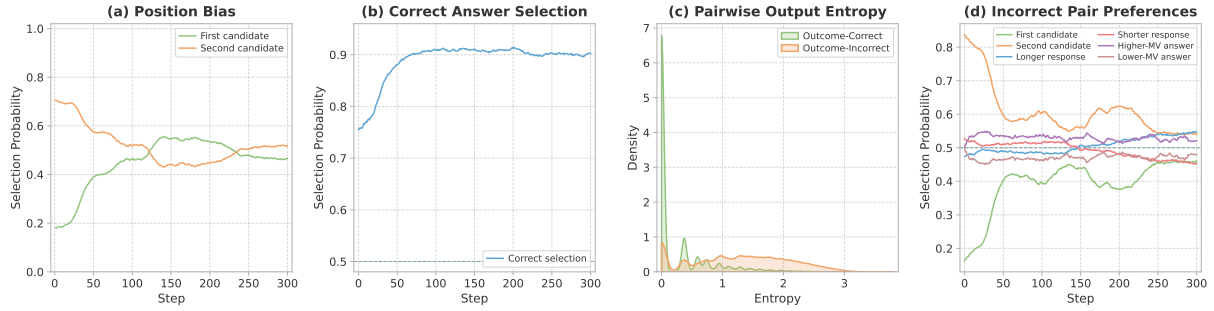


Figure 7: Mechanistic analysis of CAR. CAR improves reward estimation by amplifying reliable pairwise signals from correct reasoning trajectories while introducing no systematic bias when no correct answer exists.

CAR $3.64\times$ faster; despite this higher cost, direct sampling degrades average performance. Unlike independent rollouts that can amplify spurious consensus through raw frequency, CAR introduces pairwise interactions to estimate relative trajectory quality, yielding a more accurate and efficient use of the sampling budget.

4.7 Mechanistic Analysis of CAR

To understand how CAR improves soft reward estimation, we analyze four aspects of its behavior.

Position bias. We first examine whether CAR relies on superficial presentation order in pairwise prompts. Figure 7(a) shows that the model initially exhibits a strong preference for the second candidate. As OM-GRPO training progresses, this bias gradually diminishes and the selection probabilities of the two candidates move toward a balanced regime. Because the pairwise data are constructed symmetrically, this trend suggests that the learned comparison signal becomes increasingly insensitive to candidate order.

Correct answer selection. Figure 7(b) plots the probability that CAR selects the correct answer when a correct answer is present in the candidate set. The probability quickly rises during early training and stabilizes around 90%. This indicates that CAR reliably identifies correct answers when they appear. This behavior aligns with the role of CAR in our framework: it refines soft reward estimation by injecting inexpensive pairwise comparison signals into the augmented answer pool.

Trajectory entropy. We further analyze the entropy of pairwise outputs associated with each anchor trajectory, i.e., the uncertainty of answers produced when a trajectory is compared against others in the same group. Figure 7(c) shows a clear

separation between outcome-correct and outcome-incorrect trajectories. Correct trajectories produce a highly concentrated distribution with near-zero entropy, indicating that pairwise comparisons consistently reproduce the same answer. In contrast, incorrect trajectories exhibit substantially higher and more dispersed entropy. This entropy gap suggests that correct reasoning induces stable comparative signals, whereas incorrect reasoning produces inconsistent outcomes, allowing CAR to downweight spurious trajectories during reward estimation.

Incorrect-pair preference. Finally, we examine pairs in which both candidate answers are incorrect and different. Figure 7(d) shows that when no correct answer is present, the selection probabilities of the two candidates are close to 0.5, suggesting that CAR does not introduce systematic biases.

Overall, these results suggest that CAR improves reward estimation primarily through informative pairwise comparisons, rather than by relying on heuristic shortcuts such as favoring familiar but incorrect reasoning patterns.

5 Conclusion

In this work, we presented OM-GRPO, a label-free RLVR framework that mitigates the collapse of voting-based training by decoupling reward estimation from policy optimization. By shifting learning away from answer tokens and toward reasoning trajectories, OM-GRPO improves both robustness and performance. We further introduce CAR, a low-cost strategy that strengthens soft reward reliability through pairwise contrasts without additional rollouts. Across diverse reasoning benchmarks and three LLM backbones, OM-GRPO consistently outperforms prior label-free methods and achieves performance comparable to supervised GT-Reward training with stable, non-collapsing dynamics.

Limitations

While OM-GRPO is effective on the reasoning tasks studied in this work, several limitations remain. First, OM-GRPO requires a clear boundary between the reasoning trajectory and the final answer span, such as `\boxed{}` or `<answer>...</answer>`, together with reliable answer extraction, normalization, or verification. It is therefore most directly applicable to structured label-free RLVR; extending it to fully free-form generation, where outcomes cannot be cleanly isolated, remains an open problem. Second, our main evaluations cover open models up to 14B; broader validation at larger scales remains limited. Third, although CAR is more efficient than increasing rollout counts, it still introduces additional inference overhead, leaving room for further optimization. Finally, OM-GRPO reduces shortcut optimization on answer tokens, but it still relies on outcome-based reward signals rather than explicit supervision of reasoning quality. Incorporating richer process-level signals without reintroducing heavy annotation remains an interesting direction for future work.

Acknowledgments

This work is supported by the National Science and Technology Major Project (Grant No. 2022ZD0116101), the Project of Research and Development for Neural Machine Translation Models between Cantonese and Mandarin (Grant No. WT135-76), the Major Scientific Research Project of the State Language Commission in the 13th Five-Year Plan (Grant No. WT135-38), and the Public Technology Service Platform Project of Xiamen City (Grant No. 3502Z20231043).

References

- Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. [The unreasonable effectiveness of entropy minimization in llm reasoning](#). *CoRR*, abs/2505.15134.
- Xinjie Chen, Biao Fu, Jing Wu, Guoxin Chen, Xinggao Liu, Dayiheng Liu, and Minpeng Liao. 2026a. [Jury-rl: Votes propose, proofs dispose for label-free rlvr](#). *Preprint*, arXiv:2604.25419.
- Xinjie Chen, Minpeng Liao, Guoxin Chen, Chengxi Li, Biao Fu, Kai Fan, and Xinggao Liu. 2026b. [From data-centric to sample-centric: Enhancing LLM reasoning via progressive optimization](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2227–2242, San Diego, California, United States. Association for Computational Linguistics.
- Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. 2025. [Stop summation: Min-form credit assignment is all process reward model needs for reasoning](#). *Preprint*, arXiv:2504.15275.
- 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. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I. Wang. 2024. [Cruxeval: A benchmark for code reasoning, understanding and execution](#). *CoRR*, abs/2401.03065.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. [Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model](#). *CoRR*, abs/2503.24290.
- Hugging Face H4. 2024. [AIME 2024 Benchmark](#). https://huggingface.co/datasets/HuggingFaceH4/aime_2024.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helvar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 242 others. 2024. [Openai o1 system card](#). *Preprint*, arXiv:2412.16720.

- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. [Live-codebench: Holistic and contamination free evaluation of large language models for code](#). *CoRR*, abs/2403.07974.
- Kimi-Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. [Kimi k1.5: Scaling reinforcement learning with llms](#). *CoRR*, abs/2501.12599.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. [Tulu 3: Pushing frontiers in open language model post-training](#). *CoRR*, abs/2411.15124.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. [Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback](#). In *Proceedings of the 41st International Conference on Machine Learning*, pages 26874–26901. PMLR.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). *Preprint*, arXiv:2305.20050.
- math-ai Team. 2024. Amc23: American mathematics competitions 2023 test set. <https://huggingface.co/datasets/math-ai/amc23>. Hugging Face dataset; 40 AMC 2023 problems; accessed 2025-09-16.
- MathArena. 2025. Hmmt february 2025. https://huggingface.co/datasets/MathArena/hmmt_feb_2025.
- MathArena. 2026a. Aime 2026. https://huggingface.co/datasets/MathArena/aime_2026.
- MathArena. 2026b. Hmmt february 2026. https://huggingface.co/datasets/MathArena/hmmt_feb_2026.
- OpenCompass. 2025. AIME 2025 Benchmark. <https://huggingface.co/datasets/opencompass/AIME2025>.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). *CoRR*, abs/2203.02155.
- Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2023. [Language model self-improvement by reinforcement learning contemplation](#). *CoRR*, abs/2305.14483.
- Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. 2025. [Maximizing confidence alone improves reasoning](#). *CoRR*, abs/2505.22660.
- Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. [ReCEval: Evaluating reasoning chains via correctness and informativeness](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore. Association for Computational Linguistics.
- 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](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. 2025. [Can large reasoning models self-train?](#) *CoRR*, abs/2505.21444.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. [Hybridflow: A flexible and efficient RLHF framework](#). In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.
- Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. [Crossing the reward bridge: Expanding RL with verifiable rewards across diverse domains](#). *CoRR*, abs/2503.23829.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *International Conference on Learning Representations*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). *CoRR*, abs/2406.01574.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). *CoRR*, abs/2201.11903.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Haoran Wei, Hao Ge, Haoran Wei, Huan Lin, and others. 2025a. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- 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, and 24 others. 2025b. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaozhong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025a. [Dapo: An open-source llm reinforcement learning system at scale](#). *Preprint*, arXiv:2503.14476.
- Zhaoning Yu, Will Su, Leitian Tao, Haozhu Wang, Aashu Singh, Hanchao Yu, Jianyu Wang, Hongyang Gao, Weizhe Yuan, Jason Weston, Ping Yu, and Jing Xu. 2025b. [Restraining: From spurious votes to signals – self-driven rl with self-penalization](#). *Preprint*, arXiv:2510.02172.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?](#) *Preprint*, arXiv:2504.13837.
- Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Zhanke Zhou, Xuan Li, Xiao Feng, Jiangchao Yao, and Bo Han. 2025. [Co-rewarding: Stable self-supervised rl for eliciting reasoning in large language models](#). *arXiv preprint arXiv:2508.00410*.
- Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025a. [Absolute zero: Reinforced self-play reasoning with zero data](#). *CoRR*, abs/2505.03335.
- Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2025b. [Learning to reason without external rewards](#). *CoRR*, abs/2505.19590.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *CoRR*, abs/2311.07911.
- Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. 2025. [Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms](#). *Preprint*, arXiv:2506.18896.

A Benchmark and Metric Details

This section summarizes the evaluation protocol for each benchmark in the main text.

- **AIME24/25** (Hugging Face H4, 2024; OpenCompass, 2025). We report **avg@16** accuracy, obtained by sampling 16 independent solutions per problem and averaging correctness. Evaluation is performed with `lighteval`.⁴
- **MATH500 and GSM8K** (Lightman et al., 2023; Cobbe et al., 2021). We report **avg@4** accuracy for both benchmarks, computed by averaging correctness over 4 independently sampled solutions. Evaluation is performed with `lighteval`.⁵
- **AMC** (math-ai Team, 2024). We report **avg@8** accuracy, i.e., averaged correctness over 8 sampled solutions per problem. We use the `ttrl` evaluation implementation.⁶
- **LiveCodeBench** (Jain et al., 2024). We report **avg@5** accuracy, computed over 5 sampled solutions per instance, using the official LiveCodeBench evaluation library.⁷
- **CRUX** (Gu et al., 2024). We report **avg@5** accuracy with evaluation conducted via the ZeroEval framework.⁸

⁴<https://github.com/huggingface/lighteval>

⁵<https://github.com/huggingface/lighteval>

⁶https://github.com/ruixin31/Spurious_Rewards/tree/main/code/ttrl

⁷<https://github.com/LiveCodeBench/LiveCodeBench>

⁸<https://github.com/WildEval/ZeroEval>

Prompt

```
<|im_start|>system
Let's think step by step and output the final
answer within \boxed{ }.

<|im_start|>user
{Question}<|im_end|>

<|im_start|>assistant
Here are two possible solutions:

[Solution 1]
{Solution_1}

[Solution 2]
{Solution_2}

Based on the reasoning in the solutions above,
the correct final answer is \boxed{ }
```

Figure 8: Prompt template for CAR.

- **IFEval and MMLU-Pro** (Zhou et al., 2023; Wang et al., 2024). We report **pass@1** accuracy for both benchmarks. Evaluation is run with `lm-evaluation-harness`.⁹

B Implementation Details.

For each GRPO (Shao et al., 2024) update, we sample a batch of 128 questions and generate $G=8$ rollouts per question. We set the maximum prompt length to 512 and maximum response length to 3,072. Training is run for 6 epochs. We use the AdamW optimizer with a learning rate of 3×10^{-6} , $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$, together with a cosine learning-rate schedule and a warmup ratio of 0.1. We apply KL regularization with coefficient $\beta = 0.005$ and adopt a clipping ratio of $\epsilon = 0.2$. For generation, we sample with temperature 1.0 during training; at evaluation we use temperature 0.8 and $\text{top-}p=0.95$. For the test-time training experiments, we follow the same hyperparameters and generation settings as above, except that training is run for 30 epochs. To ensure fair and reproducible comparisons, we keep prompting consistent across methods by using the officially released chat-style templates for each model. All methods are implemented in ver1 (Sheng et al., 2025) and trained on $8 \times$ NVIDIA A100 GPUs.

⁹<https://github.com/EleutherAI/lm-evaluation-harness>

C Baseline Details

We provide detailed descriptions of all compared baselines.

- **Ground Truth (GT-Reward).** This is a supervised oracle baseline that uses human-annotated answers as rewards. For each rollout, we extract its final answer span via `ans(·)`. The reward is binary: a rollout receives 1 if `ans( $y_i$ )` exactly matches the ground-truth answer and 0 otherwise.
- **Majority Voting (MV).** This baseline uses self-consistency as a proxy for correctness (Shafayat et al., 2025). For each problem, we sample G rollouts and extract answers $\{\text{ans}(y_1), \dots, \text{ans}(y_G)\}$. Let z_{maj} denote the most frequent answer in the group. Each rollout y_i receives reward 1 if `ans( $y_i$ ) =  $z_{\text{maj}}$` and 0 otherwise. This reward reinforces agreement within the group, regardless of whether the agreed answer is correct.
- **Self-Certainty.** This baseline uses the model’s own confidence as a reward signal at the sampled trajectory (Zhao et al., 2025b). Concretely, it assigns higher reward to rollouts that the policy itself assigns higher likelihood over the entire generated sequence. Self-Certainty therefore encourages the policy to reinforce trajectories it considers more probable, without requiring any external labels.
- **Entropy Minimization.** This baseline encourages low-entropy predictions on the final answer span (Prabhudesai et al., 2025). The reward is defined as the negative (or inverse) of the policy’s token-level entropy over the extracted answer span, so that rollouts with more deterministic answer-token distributions receive higher reward.
- **CoReward.** CoReward encourages agreement under semantically equivalent input perturbations (Zhang et al., 2025). For each training question, the method constructs paraphrased variants that preserve semantics. It then samples multiple rollouts for both x and $\tilde{x}$, aggregates their extracted answers via majority voting to obtain pseudo-consensus labels, and uses these labels in a cross-over manner as rewards when optimizing the policy on both the original and paraphrased inputs.

D Theoretical Analysis

We present a theoretical analysis of label-free GRPO under majority-voting rewards. We first show that majority voting induces an inherent global mode collapse through answer-level positive feedback. We then demonstrate that this failure mode can be provably eliminated by applying the proposed Outcome-Masked Update (OMU), which blocks gradient propagation on the answer span and redirects credit assignment to the reasoning process. Importantly, the two analyses are conducted under the same majority-voting reward; the only difference lies in whether OMU is applied.

D.1 Global Mode Collapse in Label-Free GRPO with Majority Voting

We begin by analyzing the optimization dynamics of label-free GRPO when majority voting (MV) is used as the sole correctness signal. Under this setting, we show that answer-level positive feedback, combined with parameter sharing, inevitably drives the system toward a globally shared collapsed answer, independent of the input. This analysis characterizes the failure mode of vanilla MV-based training and serves as a reference point for the OMU introduced in the following subsection.

D.1.1 Problem Formulation

Let $\mathcal{X}$ denote the input space and $\mathcal{V}$ the vocabulary. For each input $x \in \mathcal{X}$, the policy π_θ generates a trajectory

$$\tau = (y, z),$$

where $y \in \mathcal{V}^T$ is a chain-of-thought reasoning sequence and $z \in \mathcal{V}^*$ denotes the final extracted answer span. We focus on the induced distribution over answer outcomes $\pi_\theta(z \mid x)$.

Answer Logits. Although the answer z may consist of multiple tokens, majority voting operates on the discrete extracted answer string. Accordingly, we model the answer distribution as categorical over possible outcomes. At the answer position, logits are parameterized as

$$\ell_k(x) = \mathbf{w}_k^\top \mathbf{h}(x) + b_k, \quad k \in \mathcal{V}, \quad (5)$$

where $\mathbf{h}(x)$ is the hidden state at the answer position, and $(\mathbf{w}_k, b_k)$ are globally shared parameters. The answer distribution is given by the softmax

$$\pi_\theta(k \mid x) = \frac{\exp(\ell_k(x))}{\sum_{j \in \mathcal{V}} \exp(\ell_j(x))}. \quad (6)$$

Group Sampling and Majority Reward. For each input x , GRPO samples a group of G trajectories

$$\{\tau_i\}_{i=1}^G \sim \pi_\theta(\cdot \mid x), \quad \tau_i = (y_i, z_i),$$

and extracts the corresponding answer spans $\{z_i\}_{i=1}^G$. Let

$$m = \text{Mode}(\{z_i\}_{i=1}^G), \quad (7)$$

with odd G or a fixed tie-breaking rule. The label-free majority-voting reward is

$$r_i = \mathbb{I}\{z_i = m\}. \quad (8)$$

Define the group mean and standard deviation

$$\mu = \frac{1}{G} \sum_{i=1}^G r_i, \quad \sigma = \sqrt{\frac{1}{G} \sum_{i=1}^G (r_i - \mu)^2}. \quad (9)$$

The GRPO advantage is

$$A_i = \frac{r_i - \mu}{\sigma + \varepsilon}, \quad (10)$$

where $\varepsilon > 0$ stabilizes the computation. By construction, $\sum_{i=1}^G A_i = 0$.

Objective. The GRPO loss is

$$\mathcal{L}(\theta) = -\mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G A_i \log \pi_\theta(z_i \mid x) \right] + \beta \mathcal{R}_{\text{KL}}(\theta). \quad (11)$$

D.1.2 Positive Update for the Group Mode

Lemma 1. Fix an input x and a sampled group $\{z_i\}_{i=1}^G$ with advantages $\{A_i\}$. Let $k^* = m$ be the group mode. Ignoring the KL term (or when the policy-gradient term dominates), the policy-gradient update satisfies $\Delta b_{k^*} > 0$.

Proof. The policy-gradient term for this group is

$$\mathcal{L}_{\text{PG}} = -\frac{1}{G} \sum_{i=1}^G A_i \log \pi_\theta(z_i \mid x). \quad (12)$$

Using the softmax identity

$$\frac{\partial}{\partial b_k} \log \pi_\theta(z_i \mid x) = \mathbb{I}\{z_i = k\} - \pi_\theta(k \mid x), \quad (13)$$

we obtain

$$\frac{\partial \mathcal{L}_{\text{PG}}}{\partial b_k} = -\frac{1}{G} \sum_{i=1}^G A_i (\mathbb{I}\{z_i = k\} - \pi_\theta(k | x)) \quad (14)$$

$$= -\frac{1}{G} \left(\sum_{i:z_i=k} A_i - \pi_\theta(k | x) \sum_{i=1}^G A_i \right). \quad (15)$$

Since $\sum_{i=1}^G A_i = 0$, this simplifies to

$$\frac{\partial \mathcal{L}_{\text{PG}}}{\partial b_k} = -\frac{1}{G} \sum_{i:z_i=k} A_i. \quad (16)$$

For $k = k^*$, all terms correspond to winners, hence $A_i > 0$ (when $\sigma > 0$), implying $\partial \mathcal{L}_{\text{PG}} / \partial b_{k^*} < 0$ and therefore $\Delta b_{k^*} > 0$ under gradient descent.

D.1.3 Global Positive Feedback via Parameter Sharing

Lemma 2. *If an outcome k^* becomes the group mode more frequently than others in a batch, resulting in a net increase $\Delta b_{k^*} > 0$, then to first order (holding other logits fixed), $\pi_\theta(k^* | x)$ increases for all x , further increasing the probability that k^* becomes the group mode in subsequent updates.*

Proof. The bias b_{k^*} is shared across all inputs. An increase Δb_{k^*} adds a constant to the logit $\ell_{k^*}(x)$ for every x . Holding other logits fixed, the softmax probability $\pi_\theta(k^* | x)$ is strictly increasing in $\ell_{k^*}(x)$. Since the probability of being the majority among G samples is a monotone function of $\pi_\theta(k^* | x)$, the frequency with which k^* becomes the group mode increases, closing a positive feedback loop.

D.1.4 Absorbing and Stable Collapsed State

Lemma 3. *The fully collapsed answer policy $\pi_\theta(z | x) = \delta(z - k^*)$ is an absorbing state and is locally stable under GRPO dynamics.*

Proof. If $\pi_\theta(z | x) = \delta(z - k^*)$, then every sampled group consists entirely of k^* , so $r_i \equiv 1$, $\mu = 1$, and $\sigma = 0$. In standard GRPO implementations, this yields $A_i = 0$ for all i , hence $\nabla_\theta \mathcal{L}_{\text{PG}} = 0$ and the parameters stop updating, making the state absorbing.

For local stability, consider a near-collapsed regime where most groups still have mode k^* , but

rare samples produce $k \neq k^*$. Such samples are necessarily losers with $A_i < 0$. From Lemma 1,

$$\frac{\partial \mathcal{L}_{\text{PG}}}{\partial b_k} = -\frac{1}{G} \sum_{i:z_i=k} A_i > 0, \quad (17)$$

implying $\Delta b_k < 0$ for $k \neq k^*$. Thus deviations are suppressed, establishing local stability.

D.1.5 Answer-Only Collapse under Tokenwise KL Regularization

Lemma 4. *Under majority-voting rewards and the tokenwise KL penalty used in GRPO, defined by*

$$\phi(r) = r - \log r - 1, \quad r_{i,t} = \frac{\pi_{\text{ref}}(\tau_{i,t} | x, \tau_{i,<t})}{\pi_\theta(\tau_{i,t} | x, \tau_{i,<t})}, \quad (18)$$

the optimization favors concentrating distributional shift on the final answer span rather than across the entire reasoning sequence.

Proof. The surrogate satisfies $\phi(r) \geq 0$ for all $r > 0$, with equality if and only if $r = 1$. The corresponding tokenwise KL regularizer takes the form

$$\mathcal{R}_{\text{KL}}(\theta) = \mathbb{E} \left[\sum_{t=1}^{|\tau_i|} \phi(r_{i,t}) \right], \quad (19)$$

where $\tau_i = (y_i, z_i)$ and the sum ranges over all tokens in the trajectory, including both the reasoning tokens in y_i and the answer-span tokens in z_i . Any deviation from the reference policy at step t incurs a nonnegative cost. Deviating across multiple reasoning steps accumulates penalties across those steps. In contrast, deviating primarily on the answer span accumulates the penalty only over the answer-span positions. Since the majority-voting reward depends only on answer agreement, collapsing the answer distribution yields large reward gains at relatively small KL cost, whereas collapsing the entire reasoning sequence incurs a much larger KL penalty.

D.1.6 Proof of Global Mode Collapse

Theorem 1. *Under label-free Group Relative Policy Optimization (GRPO) with majority-voting rewards, the answer distribution admits a globally shared collapsed mode as an attracting equilibrium. Specifically, under standard training conditions (odd group size or fixed tie-breaking, nonzero exploration, and a KL coefficient β that does not dominate the policy-gradient signal), the optimization dynamics exhibit: (i) global positive feedback*

favoring the current majority outcome due to parameter sharing, and (ii) an absorbing and locally stable collapsed answer policy.

Proof. Lemma 1 shows that any outcome becoming the group mode is reinforced independent of correctness. Lemma 2 implies that parameter sharing turns early random advantages into a global positive-feedback loop. Lemma 4 explains why optimization pressure concentrates collapse on the answer span. Finally, Lemma 3 establishes that the collapsed answer policy is absorbing and locally stable. Together, these results show that global mode collapse of the answer distribution emerges as an attracting equilibrium under label-free GRPO with majority voting.

D.2 Preventing Global Mode Collapse via OMU

We now analyze the optimization dynamics under the same MV reward, but with gradients on the answer span masked (ours OMU). We show that this modification prevents global mode collapse while preserving meaningful learning signals for the reasoning process.

D.2.1 Problem Setup

Given an input $x \in \mathcal{X}$, the policy π_θ generates a trajectory

$$\tau = (y, z),$$

where $y = (\tau_1, \dots, \tau_T)$ is the reasoning sequence and $z = (\tau_{T+1}, \dots, \tau_{T+L})$ is the final answer span, with $L \geq 1$. The evaluation unit for majority voting is the entire answer span z .

Majority-Voting Reward. For each input x , we sample a group of G trajectories $\{\tau_i\}_{i=1}^G$ with corresponding answer spans $\{z_i\}_{i=1}^G$. Let

$$m = \text{Mode}(\{z_i\}_{i=1}^G)$$

denote the most frequent answer in the group. The reward is defined as

$$r_i = \mathbb{I}\{z_i = m\}.$$

We construct a centered GRPO advantage

$$A_i = \frac{r_i - \mu}{\sigma + \varepsilon}, \quad \mu = \frac{1}{G} \sum_{i=1}^G r_i,$$

which satisfies $\sum_{i=1}^G A_i = 0$.

D.2.2 Answer-Span Gradient Masking

We introduce a binary mask over token positions for trajectory τ_i :

$$m_{i,t} = \begin{cases} 1, & 1 \leq t \leq T_i \quad (\text{reasoning tokens}), \\ 0, & T_i < t \leq |\tau_i| \quad (\text{answer span}). \end{cases}$$

The masked GRPO objective is defined as

$$\begin{aligned} \mathcal{L}_{\text{mask}}(\theta) = & -\mathbb{E}_{x \sim \mathcal{D}, \{\tau_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\tau_i|} \right. \\ & m_{i,t} A_i \log \pi_\theta(\tau_{i,t} \mid x, \tau_{i,<t}) \\ & \left. + \beta \mathbb{E}_{x \sim \mathcal{D}, \{\tau_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\tau_i|} \right. \right. \\ & \left. \left. m_{i,t} \phi(r_{i,t}) \right] \right]. \end{aligned} \quad (20)$$

where

$$\phi(r) = r - \log r - 1, \quad r_{i,t} = \frac{\pi_{\text{ref}}(\tau_{i,t} \mid x, \tau_{i,<t})}{\pi_\theta(\tau_{i,t} \mid x, \tau_{i,<t})}.$$

Importantly, both the policy-gradient term and the KL regularization term are masked on the answer span via the token-level mask $m_{i,t}$.

D.2.3 Elimination of Direct Answer-Level Reinforcement

Lemma 5. Under the masked objective (20), for any sample i and any position t such that $m_{i,t} = 0$,

$$\frac{\partial \mathcal{L}_{\text{mask}}}{\partial \log \pi_\theta(\tau_{i,t} \mid x, \tau_{i,<t})} = 0.$$

Proof. By construction, neither the policy-gradient term nor the KL regularization term in (20) contains any factor involving $\log \pi_\theta(\tau_{i,t} \mid x, \tau_{i,<t})$ for $t > T$. Therefore, the loss function is independent of all answer-span token probabilities, and their gradients vanish identically.

Implication. Lemma 5 shows that the majority-voting advantage A_i cannot directly reinforce or suppress any answer-span token. This removes the low-cost winner-takes-all shortcut that previously drove global mode collapse at the answer level.

D.2.4 Credit Assignment Redirection to Reasoning

Lemma 6. Although the majority-voting reward is defined over answer spans, all advantage-weighted learning signals are backpropagated exclusively through the reasoning tokens. Consequently, any improvement in answer consistency

must be achieved by modifying the reasoning process.

Proof. From (20), the policy-gradient term becomes

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{mask}} = & - \mathbb{E}_{x \sim \mathcal{D}, \{\tau_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\tau_i|} \right. \\ & \left. m_{i,t} A_i \nabla_{\theta} \log \pi_{\theta}(\tau_{i,t} \mid x, \tau_{i,<t}) \right] \\ & + \nabla_{\theta} \mathcal{R}_{\text{KL}}(\theta). \end{aligned} \quad (21)$$

The advantage A_i , determined solely by whether z_i equals the group mode, weights the log-probabilities of reasoning tokens only. Thus, winning trajectories ($A_i > 0$) reinforce their entire reasoning sequences, while losing trajectories ($A_i < 0$) are suppressed at the reasoning level.

Dependency Clarification. While $\nabla_{\theta} \log \pi_{\theta}(z_i)$ is masked, the answer distribution $\pi_{\theta}(z \mid y)$ still depends on θ through the final hidden state h_T , where $h_T = \text{Transformer}(y; \theta)$. Since y is optimized via the masked policy gradient, the effective optimization pathway becomes

$$\theta \rightarrow y \rightarrow h_T \rightarrow z,$$

while the direct shortcut $\theta \rightarrow z$ is eliminated.

D.2.5 KL Barrier Against Reasoning Collapse

Lemma 7. *Under the masked objective (20), any strategy that attempts to enforce answer consistency by collapsing the reasoning process incurs a KL cost that grows at least linearly with the reasoning length T .*

Proof. The KL regularization term is a sum over reasoning positions:

$$\mathcal{R}_{\text{KL}}(\theta) = \mathbb{E} \left[\sum_{t=1}^T \phi(r_{i,t}) \right],$$

where $\phi(r) \geq 0$ for all $r > 0$, with equality if and only if $r = 1$. If a collapsed reasoning strategy deviates from the reference policy on a set $\mathcal{S} \subseteq \{1, \dots, T\}$ such that $\phi(r_{i,t}) \geq c > 0$ for all $t \in \mathcal{S}$, then

$$\sum_{t=1}^T \phi(r_{i,t}) \geq \sum_{t \in \mathcal{S}} \phi(r_{i,t}) \geq |\mathcal{S}| \cdot c.$$

When reasoning collapse implies systematic deviation on a constant fraction of positions, we have $|\mathcal{S}| = \Theta(T)$, yielding a KL cost of $\Omega(T)$.

D.2.6 Prevention of Global Mode Collapse

Theorem 2. *Under label-free GRPO with majority-voting rewards, masking both policy-gradient and KL terms on the answer span prevents global mode collapse and forces learning to proceed through improvements in the reasoning process.*

Proof. Lemma 5 eliminates direct answer-level reinforcement, removing the low-cost shortcut responsible for global mode collapse. Lemma 6 shows that all reward-induced learning signals are redirected to the reasoning process. Lemma 7 establishes that enforcing answer consistency via reasoning collapse incurs a KL cost that scales linearly with sequence length, rendering such strategies unfavorable. Therefore, the only viable optimization path is to improve reasoning behaviors that reliably support correct answers.

E Additional Analysis

E.1 Low-Contamination Benchmark Results

To further examine whether OM-GRPO’s gains depend on benchmark exposure in pretraining, we evaluate Qwen2.5-7B on three recently released math benchmarks whose public releases postdate Qwen2.5-7B: AIME26 (MathArena, 2026a), HMMT25 (MathArena, 2025), and HMMT26 (MathArena, 2026b). As shown in Table 5, OM-GRPO achieves the highest average Pass@16 among all compared methods and the second-best average Avg@16, behind only the supervised GT-Reward baseline.

E.2 Pass@k Results

Table 6 reports Pass@k performance across three backbones. OM-GRPO consistently outperforms other label-free baselines and remains competitive with GT-Reward, indicating better correct-solution coverage. This result is consistent with the trends in Figure 9, which show that OM-GRPO also preserves answer diversity throughout training.

E.3 Answer Diversity during Training

A complementary view of training stability is provided by answer diversity. Figure 9 shows that Majority Voting suffers from severe answer collapse: the average number of unique answers steadily drops and eventually approaches one, indicating convergence to a single answer per problem. In contrast, OM-GRPO (with or without CAR) maintains a substantially higher and more stable level

Methods	AIME26		HMMT25		HMMT26		Average	
	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16
GT-Reward	6.46 $\pm$ 1.37	23.33	1.67 $\pm$ 0.92	10.00	4.36 $\pm$ 1.02	12.12	4.16	15.15
Self-Certainty	3.12 $\pm$ 1.21	16.67	0.00 $\pm$ 0.00	10.00	2.46 $\pm$ 1.21	12.12	1.86	12.93
Entropy	3.96 $\pm$ 1.33	20.00	0.42 $\pm$ 0.61	13.33	0.38 $\pm$ 0.55	9.09	1.58	14.14
Majority Voting	3.54 $\pm$ 1.52	10.00	0.00 $\pm$ 0.00	3.33	3.03 $\pm$ 0.00	6.06	2.19	6.46
CoReward	3.75 $\pm$ 1.43	16.67	0.21 $\pm$ 0.44	3.33	0.00 $\pm$ 0.00	9.09	1.32	9.70
OM-GRPO	4.58 $\pm$ 1.28	20.00	0.62 $\pm$ 0.72	13.33	4.55 $\pm$ 0.83	21.21	3.25	18.18

Table 5: Low-contamination benchmark results (%) on Qwen2.5-7B. AIME26, HMMT25, and HMMT26 are recently released benchmarks whose public releases postdate Qwen2.5-7B.

Methods	AIME24 Pass@16	AIME25 Pass@16	MATH500 Pass@4	GSM8K Pass@4	AMC Pass@8	Average
<i>Qwen3-1.7B-Base</i>						
Before RL	16.67	13.33	72.0	89.31	56.63	49.59
GT-Reward	26.67	16.67	82.0	92.42	59.04	55.36
Entropy Minimization	20.00	23.33	81.2	90.75	59.04	54.86
Self-Certainty	23.33	26.67	77.2	90.83	57.83	55.17
Majority Voting	23.33	16.67	77.2	92.34	53.01	52.51
CoReward	26.67	16.67	77.4	92.04	54.22	53.40
OM-GRPO	20.00	26.67	79.4	92.49	59.04	55.52
<i>Llama-3.2-3B-Instruct</i>						
Before RL	23.33	6.67	65.0	86.88	49.40	46.26
GT-Reward	23.33	6.67	63.8	90.14	43.37	45.46
Entropy Minimization	16.67	3.33	58.8	84.46	50.60	42.77
Self-Certainty	13.33	10.00	58.0	84.69	44.58	42.12
Majority Voting	16.67	0.00	63.6	90.98	39.76	42.20
CoReward	23.33	0.00	60.4	90.45	43.37	43.51
OM-GRPO	23.33	3.33	67.2	89.61	48.19	46.33
<i>Qwen2.5-7B</i>						
Before RL	30.00	20.00	76.2	93.78	63.86	56.77
GT-Reward	33.33	30.00	85.2	95.22	68.67	62.48
Entropy Minimization	30.00	36.67	85.2	93.78	67.47	62.62
Self-Certainty	23.33	30.00	84.6	93.86	69.88	60.33
Majority Voting	33.33	30.00	85.4	95.00	63.86	61.52
CoReward	30.00	26.67	83.8	95.60	63.86	59.99
OM-GRPO	33.33	30.00	85.2	95.45	66.27	62.05

Table 6: Pass@k Results (%) of RL performance comparison on math reasoning benchmarks.

of answer diversity throughout training, closely tracking the GT-Reward baseline. This indicates that OM-GRPO suppresses answer-level shortcuts and preserves meaningful diversity under label-free optimization.

E.4 Rollout-Level Pseudo-Label Bias Analysis

To distinguish random pseudo-label noise from policy-induced bias, we analyze rollouts from Qwen3-1.7B-Base throughout training. Figure 10 tracks top-answer share (the fraction of rollouts producing the most frequent answer), accuracy, mean training reward, and predicted-1 share. Majority Voting progressively collapses to predicting 1: over

the last 50 steps, both shares approach 1.0, although only 2.4% of ground-truth answers are 1. Meanwhile, its training reward rises from 1.354 to 1.996 as accuracy falls from 44.0% to 2.4%. This divergence shows that the policy’s frequent answer becomes the consensus pseudo-label and is then self-reinforced. OM-GRPO avoids this feedback loop by masking answer-span gradients, keeping its answer distribution and accuracy close to those of GT-Reward.

E.5 PRM-based Process Reward Baselines

We test whether process reward supervision prevents collapse using ReasonFlux-PRM-1.5B,

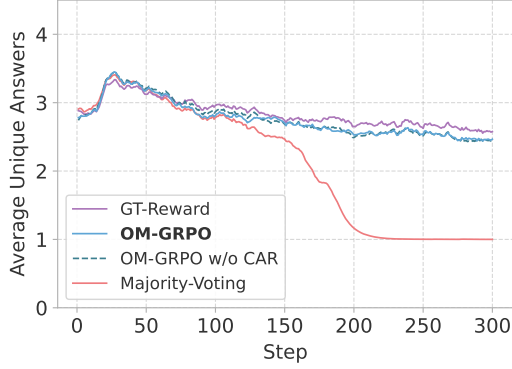


Figure 9: Average unique answers per sample over training steps on Qwen3-1.7B-Base.

Method	Correctness	Informativeness	Answer Support
MV	77.78	68.89	50.00
OM-GRPO	89.29	80.95	80.00
Δ	+11.51	+12.06	+30.00

Table 7: Reasoning-quality evaluation on AIME25 with Qwen3-1.7B-Base (%).

ReasonFlux-PRM-7B (Zou et al., 2025), and PURE-PRM-7B (Cheng et al., 2025) on Qwen3-1.7B-Base. These label-free baselines use an additional reward model to score reasoning processes.

As shown in Figure 11, all three PRM-based baselines collapse within 30–40 steps, whereas OM-GRPO improves from 53.0% to 64.4% and finishes at 63.7%. Thus, auxiliary PRM supervision alone does not prevent collapse, while OM-GRPO stabilizes training without an additional reward model.

E.6 Direct Process-Quality Evaluation

To directly evaluate reasoning quality, we compare MV and OM-GRPO on AIME25, with Qwen3-1.7B-Base as the backbone and DeepSeek-V4-Flash as the evaluator, using three ReCEval-style metrics (Prasad et al., 2023): (1) Correctness, the proportion of logically valid steps consistent with the problem and preceding reasoning; (2) Informativeness, the proportion of steps that contribute useful information toward the solution; and (3) Answer Support, the proportion of trajectories whose reasoning justifies the final answer. As shown in Table 7, OM-GRPO improves all three metrics over MV, indicating that outcome masking improves reasoning quality rather than merely stabilizing answer tokens.

E.7 Robustness to Weak Initial Reasoning Ability

To understand how exploration quality affects label-free RLVR, we analyze this issue from two complementary angles. Weak base models provide a natural test bed for reduced exploration quality, since they typically generate far fewer correct trajectories at initialization, and in some cases the initial Pass@8 can be close to zero.

As shown in Figure 12, although overall performance decreases as the base model becomes weaker, OM-GRPO remains stable across model scales. In contrast, the MV baseline becomes increasingly prone to late-stage collapse. This suggests that limited initial exploration quality does not by itself force training into a consistent but incorrect mode. We further examine an extreme regime by tracking 3,795 samples whose initial Pass@8 is exactly zero, meaning that none of the initial sampled trajectories is correct. If correct trajectories were required from the start, such samples would remain unrecoverable throughout training. However, the right panel shows that cumulative Pass@8 steadily rises to about 65%, while cumulative Avg@8 reaches about 49%, indicating that many initially failed samples are eventually solved and that overall solution quality also improves.

This behavior is consistent with the design of OM-GRPO. Because gradients on the answer span are masked, the model does not directly reinforce the majority answer string. Even when early roll-outs are consistently incorrect, updates mainly affect reasoning tokens rather than answer tokens, which helps avoid convergence to a self-consistent but incorrect answer mode.

E.8 Stability under Strong Base Initialization

We evaluate Qwen3-14B-Base over 10 epochs to test whether MV degradation stems only from weak initialization. As shown in Figure 13, MV peaks at 79.77 but falls to 75.09 by the final checkpoint, whereas OM-GRPO peaks at 80.58 and ends at 80.25. Thus, stronger initialization delays but does not eliminate MV degradation, while OM-GRPO remains stable by preventing direct optimization of answer tokens.

E.9 Generalization to Open-Ended Generation

We further evaluate OM-GRPO on the open-ended WritingBench task under a test-time training setup.

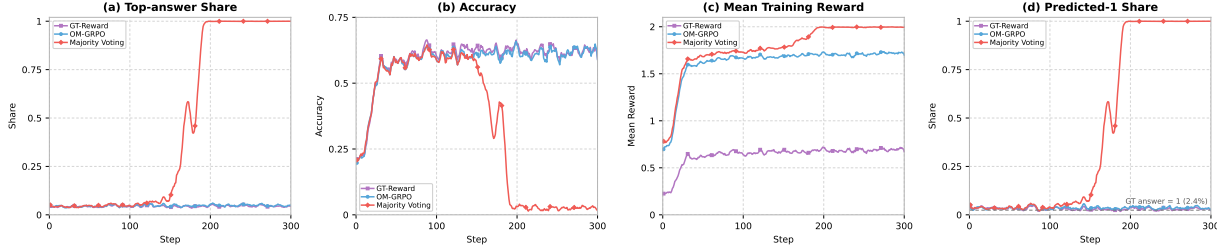


Figure 10: Rollout-level analysis on Qwen3-1.7B-Base, including top-answer share, accuracy, mean training reward, and predicted-1 share.

Model	Overall	Academic & Engineering	Finance & Business	Politics & Law	Literature & Arts	Education	Advertising
Entropy Minimization	3.52	3.22	3.37	3.05	3.64	3.97	4.32
OM-GRPO	4.04	4.01	3.74	4.10	4.03	4.27	4.32

Table 8: WritingBench benchmark results under test-time training. Scores are computed using GPT-5-nano as the judge model. Results for EM are reported from the last checkpoint before collapse, while OM-GRPO results are taken from the final checkpoint.

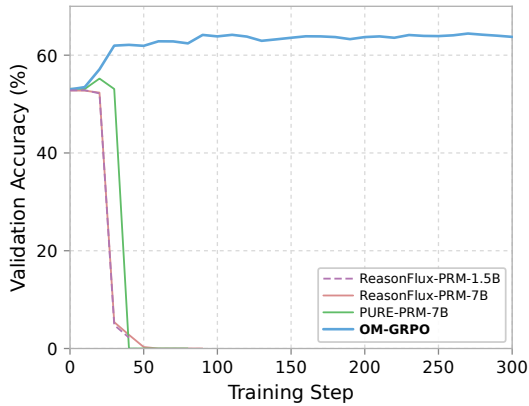


Figure 11: MATH5000 validation accuracy of PRM-based baselines and OM-GRPO on Qwen3-1.7B-Base.

In this experiment, training uses an entropy-based reward defined on the answer span, following the entropy-minimization (EM) objective of [Prabhudesai et al. \(2025\)](#). Each output follows the structured format `<think>{reasoning}</think><answer>{answer}</answer>`. We compare OM-GRPO with the EM baseline of [Prabhudesai et al. \(2025\)](#), which directly minimizes answer-span entropy without masking. In contrast, OM-GRPO masks gradients on the answer span during optimization.

Figure 14 shows that the two methods exhibit markedly different optimization behaviors. The EM baseline drops rapidly in the early stage and soon approaches zero. Examining its outputs reveals a trivial collapse pattern: for nearly

all inputs, the model generates the literal string `<answer>...</answer>`, i.e., an answer span containing only three dots. In contrast, OM-GRPO avoids this collapse. Although its score also decreases at the beginning of training, it later recovers and remains substantially higher throughout optimization. This difference is also reflected in the final WritingBench results in Table 8. OM-GRPO improves the overall score from 3.52 to 4.04 and achieves better results across all domains. These results support two observations. First, OM-GRPO extends naturally to the open-ended WritingBench task with only a simple structural constraint on the output format, and remains effective under test-time training. Second, a similar collapse pattern is also observed under this entropy-based reward, while OM-GRPO continues to deliver more stable optimization and better final performance.

F Computational Overhead

Table 9 compares OM-GRPO with Majority Voting on Qwen3-1.7B-Base using $n=8$ rollouts, with time averaged over 348 training steps and memory measured from 8-GPU logs. OM-GRPO increases per-step training time from 37.71 to 59.12 seconds (+56.8%), primarily because reward computation rises from 0.40 to 19.22 seconds. Peak GPU memory remains essentially unchanged (60.34 vs. 60.28 GiB) because OM-GRPO requires no additional reward model.

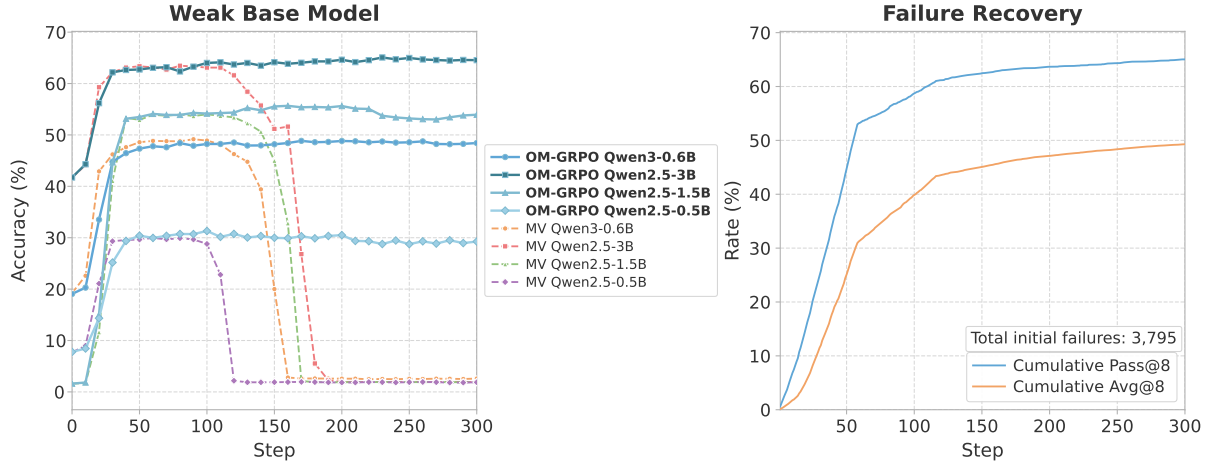


Figure 12: Effect of base-model strength on label-free RLVR.



Figure 13: Validation accuracy of Majority Voting and OM-GRPO on Qwen3-14B-Base over 10 training epochs.

<i>Training Time</i>		
Component	OM-GRPO	Majority Voting
Generation	21.91	20.14
Reward	19.22	0.40
Old-policy log prob	3.34	3.24
Reference-model log prob	3.15	3.00
Advantage	0.03	0.03
Actor update	11.48	10.91
Subtotal	59.12	37.71
<i>GPU Memory</i>		
Metric	OM-GRPO	Majority Voting
Max GPU peak	60.34 GiB	60.28 GiB
Mean GPU peak	59.49 GiB	59.31 GiB
Active mean per GPU	46.93 GiB	40.67 GiB

Table 9: Computational overhead on Qwen3-1.7B-Base with $n=8$ rollouts. Time is reported in seconds per step; memory is reported in GiB from 8-GPU training logs.

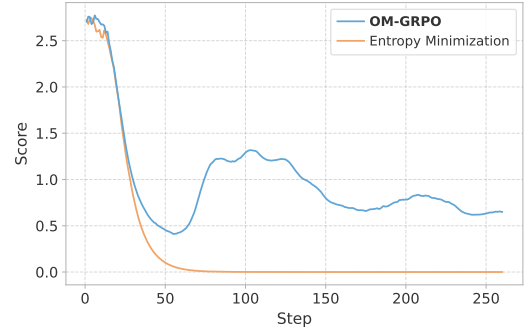


Figure 14: Test-Time Training dynamics on the open-ended WritingBench task.

G Case Study

To illustrate the reasoning–answer mismatch failure mode in label-free RLVR and the effectiveness of our proposed OM-GRPO, we analyze a representative example shown in Figure 15, which contrasts an MV baseline output with the output produced by OM-GRPO on the same algebraic problem.

Reasoning–Answer Mismatch. In the Figure 15 example, the baseline model generates a seemingly valid derivation: it correctly solves the denominator equation and identifies the two vertical asymptotes at $x = -\frac{2}{3}$ and $x = -1$, implying $a + b = -\frac{5}{3}$. However, the final answer contradicts this derivation and reports an incorrect result. This behavior highlights a structural limitation of outcome-centric optimization in label-free RLVR: the reward signal is agnostic to whether the final prediction is logically entailed by the preceding reasoning trace. As a result, the model may satisfy the reward objective via answer-level shortcuts while breaking the causal dependency between reasoning and out-

come.

Reasoning–Answer Consistency under OM-GRPO. In the Figure 16 example, the output produced by OM-GRPO exhibits a consistent reasoning–answer relationship. The model derives the asymptotes and subsequently reports the matching value of $a + b$, preserving a coherent causal chain from intermediate computation to the final answer. This improvement follows directly from OM-GRPO’s masked credit assignment: by excluding answer tokens from gradient updates, the optimization pressure is redirected toward the reasoning process, making the final prediction a consequence of the reasoning trace rather than an independently optimized outcome.

Case: Reasoning–Answer Mismatch

Question: The graph of $y = \frac{5x^2-9}{3x^2+5x+2}$ has vertical asymptotes at $x = a$ and $x = b$. Find $a + b$.

Model Output:

To find the vertical asymptotes of the function $y = \frac{5x^2-9}{3x^2+5x+2}$, we first need to determine where the denominator is zero. The vertical asymptotes occur where the denominator, $3x^2 + 5x + 2$, is equal to zero. Let's solve for x :

$$3x^2 + 5x + 2 = 0$$

We will use the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, where $a = 3$, $b = 5$, and $c = 2$. Substitute a , b , and c into the quadratic formula:

$$x = \frac{-5 \pm \sqrt{5^2 - 4 \times 3 \times 2}}{2 \times 3}$$

$$x = \frac{-5 \pm \sqrt{25 - 24}}{6}$$

$$x = \frac{-5 \pm \sqrt{1}}{6}$$

$$x = \frac{-5 \pm 1}{6}$$

This gives two solutions:

$$x = \frac{-5 + 1}{6} = \frac{-4}{6} = -\frac{2}{3}$$

$$x = \frac{-5 - 1}{6} = \frac{-6}{6} = -1$$

So, the vertical asymptotes occur at $x = -\frac{2}{3}$ and $x = -1$.

Thus, $a = -\frac{2}{3}$ and $b = -1$.

To find $a + b$:

$$a + b = -\frac{2}{3} + (-1)$$

$$a + b = -\frac{2}{3} - 1$$

$$a + b = -\frac{2}{3} - \frac{3}{3}$$

$$a + b = -\frac{5}{3}$$



Therefore, the answer is:

1



Figure 15: A Case Illustrating Reasoning–Answer Mismatch.

Case: Reasoning–Answer Consistency

Question: The graph of $y = \frac{5x^2-9}{3x^2+5x+2}$ has vertical asymptotes at $x = a$ and $x = b$. Find $a + b$.

Model Output:

To find the vertical asymptotes of the function $y = \frac{5x^2-9}{3x^2+5x+2}$, we need to determine the values of x where the denominator is zero (since the function approaches infinity or negative infinity at these points).

Let's start by solving the equation $3x^2 + 5x + 2 = 0$. First, we need to factorize the quadratic equation.

1. Factor the denominator:

$$3x^2 + 5x + 2 = 0$$

To factor this, we look for two numbers that multiply to $3 \cdot 2 = 6$ and add up to 5. These numbers are 2 and 3. Thus, we can rewrite the middle term $5x$ as $2x + 3x$:

$$3x^2 + 2x + 3x + 2 = 0$$

Now, we factor by grouping:

$$3x^2 + 2x + 3x + 2 = x(3x + 2) + 1(3x + 2) = (3x + 2)(x + 1) = 0$$

Setting each factor equal to zero gives us the potential asymptotes:

$$3x + 2 = 0 \quad \text{and} \quad x + 1 = 0$$

Solving these linear equations:

$$3x + 2 = 0 \implies 3x = -2 \implies x = -\frac{2}{3}$$

$$x + 1 = 0 \implies x = -1$$

Thus, the vertical asymptotes are at $x = -\frac{2}{3}$ and $x = -1$. The values a and b are therefore $-\frac{2}{3}$ and -1 respectively. We need to find $a + b$:

$$a + b = -\frac{2}{3} + (-1) = -\frac{2}{3} - 1 = -\frac{2}{3} - \frac{3}{3} = -\frac{5}{3}$$

✓

Thus, the final answer is:

$$\boxed{-\frac{5}{3}}$$

✓

Figure 16: A Case Demonstrating Reasoning–Answer Consistency Achieved by OM-GRPO.