You Need Reasoning to Learn Reasoning: The Limitations of Label-Free RL in Weak Base Models

Shuvendu Roy, Hossein Hajimirsadeghi, Mengyao Zhai, Golnoosh Samei
RBC Borealis
shuvendu.roy@rbc.com, {hossein.hajimirsadeghi, mengyao.zhai, golnoosh.samei}@borealisai.com

Abstract

Recent advances in large language models have demonstrated the promise of unsupervised reinforcement learning (RL) methods for enhancing reasoning capabilities without external supervision. However, the generalizability of these label-free RL approaches to smaller base models with limited reasoning capabilities remains unexplored. In this work, we systematically investigate the performance of label-free RL methods across different model sizes and reasoning strengths, from 0.5B to 7B parameters. Our empirical analysis reveals critical limitations: label-free RL is highly dependent on the base model's pre-existing reasoning capability, with performance often degrading below baseline levels for weaker models. We find that smaller models fail to generate sufficiently long or diverse chain-of-thought reasoning to enable effective self-reflection, and that training data difficulty plays a crucial role in determining success. To address these challenges, we propose a simple yet effective method for label-free RL that utilizes curriculum learning to progressively introduce harder problems during training and mask no-majority rollouts during training. Additionally, we introduce a data curation pipeline to generate samples with predefined difficulty. Our approach demonstrates consistent improvements across all model sizes and reasoning capabilities, providing a path toward more robust unsupervised RL that can bootstrap reasoning abilities in resource-constrained models.

1 Introduction

Recent advances in large language models (LLMs) have highlighted the effectiveness of reinforcement learning (RL) techniques for enhancing reasoning capabilities, particularly in domains like mathematics and code generation. However, traditional approaches such as Reinforcement Learning from Human Feedback (RLHF) and Reinforcement Learning with Verifiable Rewards (RLVR) rely heavily on external supervision, including human annotations or domain-specific ground-truth labels [1, 2]. This dependency poses significant scalability challenges, as acquiring such supervision becomes increasingly costly and impractical for emerging, complex tasks. To address this, recent works have proposed unsupervised paradigms that enable models to self-improve without labelled data. For instance, Test-Time Reinforcement Learning (TTRL) [3] leverages majority voting on unlabeled test data to estimate rewards, allowing models to adapt and evolve during inference. Similarly, Reinforcement Learning from Internal Feedback (RLIF), as exemplified by Intuitor [4], uses intrinsic signals like the model's own confidence (self-certainty) to drive optimization, eliminating the need for external verifiers.

Despite these promising developments, existing unsupervised RL approaches have primarily been evaluated on relatively large encoder-only models that already possess decent reasoning capabilities. For example, both TTRL and Intuitor focus on backbones from the Qwen series, such as Qwen2.5-Math-7B and Qwen2.5-7B, which are known for their strong baseline performance in reasoning tasks due to extensive pre-training. However, it remains unclear how these methods perform on models lacking such inherent capabilities, such as smaller LLMs or tasks where the pre-trained LLM does not have pre-existing knowledge for complex reasoning.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

In this paper, we investigate how unsupervised reinforcement learning methods adapt to smaller, pure base models in a label-free environment. Our findings reveal that these smaller models, which have weaker reasoning capabilities, struggle to learn in unlabeled settings, including both pseudo-labelling and self-consistency setups. We identify several key factors contributing to the failure of existing methods and propose a curriculum-based unsupervised RL approach. Our method demonstrates stronger generalization across different model types and sizes. Below is a summary of our main takeaways.

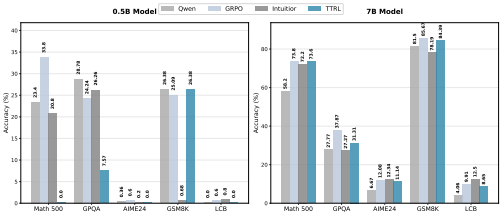

Figure 1: The performance of the Qwen2.5-0.5B base model, compared to the Qwen2.5-7B, shows that smaller models with weaker reasoning capabilities do not improve with label-free RL training.

Overview of takeaways

- Label-free RL is highly dependent on the **reasoning capability of the base model**. Performance drops significantly, sometimes worse than the base model, if the base model's reasoning ability is insufficient.
- Smaller base model (with limited reasoning) **does not generate a longer chain-of-thought** to elicit self-reflection (Aha moment).
- Length of chain-of-thought is **not a direct reflection of strong reasoning** for label-free RL.
- The **difficulty of the training data plays an important role**. A base model with limited reasoning ability cannot effectively learn from very hard problems, where it can rarely, if ever, generate the correct solution.
- Our simple training modification, which employs a **curriculum learning approach** that begins with easier problems and gradually introduces more challenging ones, enhances the performance of label-free RL across all model sizes and reasoning capabilities. Performance can be further enhanced by curating supplementary training datasets of predefined difficulty levels and employing a masked reward strategy for non-majority samples.

2 Analysis on Label-free RL

2.1 Preliminary

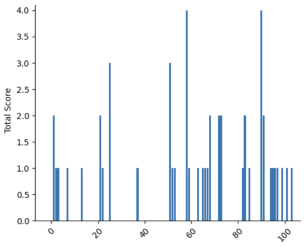

Figure 2: Correct answers generated in early stages of training by Qwen2.5-0.5B.

To establish a foundation for our analysis, we first provide an overview of the two key unsupervised RL methods: Test-Time Reinforcement Learning (TTRL) and Intuitor. TTRL [3] introduces a framework for performing RL directly on unlabeled test data during inference. Given a prompt x, the method samples multiple outputs $\{y_1, \ldots, y_N\}$ from the policy π_θ. A label y^* is estimated via majority voting on the extracted answers, and binary rewards are assigned: $r(\hat{y}_i, y^*) = 1$ if $\hat{y}_i = y^*$, else 0. The policy is then optimized to maximize the expected reward using algorithms like Group Relative Policy Optimization (GRPO). Intuitor [4] replaces external rewards with the model's intrinsic self-certainty—a measure of confidence defined as the average KL divergence from a uniform distribution over the vocabulary. Using GRPO, advantages are computed from normalized self-certainty scores, guiding the policy toward higher-confidence outputs.

2.2 Label-free RL struggles with smaller models

To investigate the effectiveness of verifier-free reinforcement learning methods on models with varying baseline capabilities, we conduct extensive experiments comparing the performance of Qwen2.5-0.5B and Qwen2.5-7B across multiple reasoning benchmarks. Our results, presented in Figure 1, reveal a striking disparity between the smaller and larger models. For the Qwen2.5-7B model, we observe consistent improvements across all evaluated benchmarks when applying verifier-free RL methods. On Math 500, the base model achieves a score of 58.2 points, which

increases to 74.6 with TTRL and 72.2 with Intuitor, comparable to the 73.8 achieved by verifier-based (using label) GRPO. Similar positive trends are evident on GPQA, AIME24, GSM8K, and LCB, with improvements ranging from modest to substantial. In stark contrast, the Qwen2.5-0.5B model exhibits fundamentally different behaviour under the same training regimes. Across all benchmarks and methods, we observe either only marginal gains or, more concerningly, performance degradation. On Math 500, the 0.5B base model achieves 23.4, but its performance declines when trained with Intuitor, and the model completely collapses when trained with TTRL. Similar patterns are observed across the other benchmarks. To further investigate the cause of model collapse, in Figure 2 we plot the number of correct rollouts at the early stages of training for the base Qwen2.5-0.5B model. As shown, the rollouts generated by the base model often contain no correct outputs. Consequently, majority voting in TTRL produces incorrect pseudo-labels. Training on these erroneous pseudo-labels ultimately leads to model collapse.

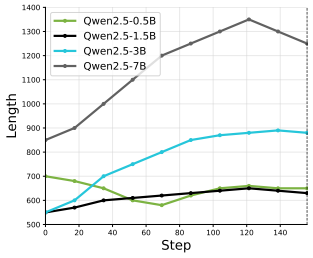

Figure 3: Comparison of average response length over the training steps.

2.3 Weaker models do not generate long chain-of-thought (CoT)

One interesting trend we observe is that the reasoning COT length for the stronger model (Qwen2.5-7B) increases as training progresses. As noted in prior work [2], longer reasoning chains often include self-reflection (the so-called "Aha moment"). In contrast, the weaker model does not exhibit such long chains. However, generation length alone is not a definitive indicator of improved reasoning. For example, Qwen 0.5B and 1.5B show similar reasoning length, even though the performance is much better for the 1.5B variant.

2.4 Difficulty of training data plays a critical role in label-free RL

To better understand the role of training data distribution, we study the effect of increasing task difficulty on the Qwen2.5-0.5B model. Figure 4 shows performance trends across Math-500, GPQA, and GSM8K as the training data shifts from relatively easier subsets (Level 1-2) to harder or less aligned subsets (Level 1-5). A clear degradation emerges as the data difficulty increases. For Math-500, the performance drops sharply, with the model nearly collapsing by Level 1-4. GPQA and GSM8K show a similar downward trend, though less steep. This suggests that weak base models are particularly sensitive to data complexity and distributional mismatch. These results highlight an important principle: choosing the right difficulty of training data is critical for effective learning. If the data is too challenging or comes from an unfamiliar distribution, weaker models may fail to generalize and instead suffer from performance degradation. Conversely, aligning training data difficulty with the model's baseline capacity appears essential for stable improvement

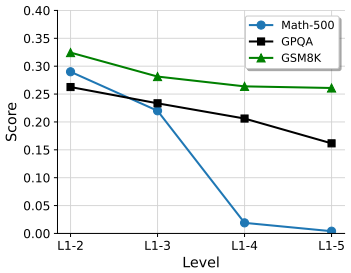

Figure 4: Correct answers generated in early stages of training by Qwen2.5-0.5B.

3 Method

To enhance the reasoning capabilities of language models while learning in a label-free setting, we propose **Cu**rriculum-guided **Ma**sked Majority Voting Reinforcement Learning (**CuMa**). This method utilizes a curriculum learning approach to guide a reinforcement learning process with a majority voting reward signal. By starting with easy samples and gradually increasing the difficulty, the model can learn the reasoning process effectively. Additionally, we introduce a data curation pipeline for generating synthetic data of predefined difficulty levels to stabilize model training and improve reasoning performance. Finally, we employ a reward-masking approach for training in the GRPO setup, where we mask the rewards for samples with no majority prediction. As motivated in Figure 2, many rollouts, especially at the start of training, generate no majority prediction. Masking the learning signal on such samples ensures that the model does not receive negative feedback from inconclusive examples.

Specifically, we partition an unlabeled dataset $\mathcal{D} = \{x_1, \ldots, x_M\}$ (e.g., math problems) into $K = 5$ difficulty bins, $\mathcal{D}_1, \ldots, \mathcal{D}_K$, where $\mathcal{D}_1$ contains the easiest prompts and $\mathcal{D}_K$ the most challenging.

Training proceeds sequentially from $\mathcal{D}_1$ to $\mathcal{D}_K$. For each bin $\mathcal{D}_k$, we sample a batch of prompts $\{x_i\}_{i=1}^{B}$ where B is the batch size. The model generates multiple candidate solutions $\{y_i^{(j)}\}_{j=1}^{N}$ for each prompt x_i, where N is the number of candidate solutions per prompt. We then apply reinforcement learning using a reward signal derived from majority voting on these solutions:

$$r(x_i, y_i^{(j)}) = \mathbb{I}[y_i^{(j)} = \text{majority\_vote}(\{y_i^{(1)}, \ldots, y_i^{(N)}\})] \quad (1)$$

where $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if the condition is true and 0 otherwise. Since small models often struggle with hard samples, this curriculum-based approach allows them to build foundational reasoning skills on easy problems first, which in turn helps them gradually learn to solve more difficult samples.

Another key component of our proposed solution is the reward masking mechanism for samples where $\max_j |\{k : y_i^{(k)} = y_i^{(j)}\}| < 2$, i.e., no majority consensus exists among the N candidates. During early training, small models generate diverse incorrect solutions, creating scenarios with no dominant answer. Rather than assigning arbitrary rewards to these inconclusive samples, we mask their learning signal entirely:

$$\text{mask}(x_i) = \mathbb{I}\left[\max_j |\{k : y_i^{(k)} = y_i^{(j)}\}| \geq 2\right] \quad (2)$$

This prevents learning from noisy feedback when predictions are uncertain, focusing the RL process on high-confidence samples while avoiding interference from ambiguous cases. The approach is particularly beneficial during initial curriculum phases when the majority consensus reliably indicates solution quality.

Method	Math 500	GPQA	AIME24	GSM8K	LCB
0.5B Models					
Base Model	23.4	28.78	0.36	26.38	0.0
GRPO	33.8	24.24	0.6	25.09	0.6
Intuitor	20.8	**26.26**	0.2	0.68	**0.8**
TTRL	0.0	7.57	0.0	26.38	0.0
Ours	**32.8**	22.72	**0.52**	**32.9**	0.2
1.5B Models					
Base Model	3.8	17.17	0.16	55.26	0.07
GRPO	57.4	23.23	3.33	58.90	2.42
Intuitor	47.0	22.22	1.4	44.57	**4.85**
TTRL	53.6	25.25	**3.85**	58.45	2.45
Ours	**54.2**	**25.75**	2.49	**59.96**	3.66
3B Models					
Base Model	54.2	30.80	3.33	73.31	5.20
GRPO	64.4	32.32	5.46	66.48	7.65
Intuitor	59.6	**30.30**	4.01	26.56	7.83
TTRL	63.8	27.27	3.33	**74.60**	7.99
Ours	**64.4**	27.27	**5.31**	72.85	**8.04**
7B Models					
Base Model	58.2	27.77	6.67	81.50	4.06
GRPO	73.8	37.87	12.08	85.67	9.91
Intuitor	72.2	27.27	12.34	78.19	**12.5**
TTRL	73.6	31.31	11.14	84.39	8.65
Ours	**74.0**	**32.32**	**13.33**	**84.49**	10.31

Table 1: Performance comparison across different model sizes and methods. Results of existing methods are reproduced in an identical training setup for fair comparison.

To facilitate further learning from easier samples before being exposed to hard samples, we have curated additional unlabelled samples by using LLM as the data generator, creating synthetic problems at varying difficulty levels to augment the training curriculum. Details on the data curation pipeline are presented in the Appendix.

4 Results

Table 2 reports the performance of our **CuMa** method compared to baselines and state-of-the-art approaches across multiple reasoning benchmarks. Overall, **CuMa** consistently improves reasoning performance, particularly on challenging tasks such as Math 500 and GSM8K, achieving substantial gains over GRPO, Intuitor, and TTRL. Our approach is effective across model scales, from 0.5B to 7B parameters, and demonstrates strong generalization across datasets. While performance gains are observed at all model sizes, improvements are more pronounced for smaller models with weaker prior reasoning capabilities. Importantly, we do not observe model collapse on any dataset or scale. We provide additional experiments and implementation details in the Appendix.

5 Conclusion

This work systematically explores the limitations of label-free reinforcement learning for enhancing reasoning in large language models, particularly smaller models with limited baseline capabilities. Our analysis reveals that existing methods struggle with smaller models due to insufficient chain-of-thought diversity and sensitivity to training data difficulty, often leading to performance degradation. To address this, we propose **CuMa**, a curriculum-guided masked majority voting RL approach that leverages progressive difficulty, curated synthetic data, and reward masking to achieve consistent improvements across model sizes (0.5B to 7B) on reasoning benchmarks.

References

[1] 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.

[2] Yi Guo, Lei Zhang, Hao Chen, Ming Li, Jun Wang, et al. Deepseek-r1: Advancing reasoning in large language models with reinforcement learning. *arXiv preprint arXiv:2501.01234*, 2025.

[3] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

[4] Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

[5] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[6] Long Ouyang, Jeff 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.

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

[8] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[9] Rong Cui, Liang He, Chen Wu, Yifan Zhang, and Qing Liu. Process-level reinforcement learning for stable reasoning in large models. *arXiv preprint arXiv:2503.04567*, 2025.

[10] Fang Yu, Zhiwei Li, Yue Wang, Hao Liu, and Jian Sun. Dapo: Direct alignment process optimization for reasoning models. *arXiv preprint arXiv:2504.05678*, 2025.

[11] Chen Liu, Ke Zhang, Rui Wang, Bo Li, and Yi Ma. Understanding the stability of reinforcement learning for reasoning. *arXiv preprint arXiv:2502.03456*, 2025.

[12] Tian Hu, Lin Zhao, Jiawei Chen, Kai Zhou, and Yan Xu. Open verification for reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.06789*, 2025.

[13] Xuezhi Wei, Tom Brown, Alice Zhou, Han Chen, and Nan Du. Swe-bench: Benchmarking verifiable reasoning in language models. *arXiv preprint arXiv:2501.07890*, 2025.

[14] Bo Yuan, Wei Zhang, Ruiqi Yang, Hang Zhou, and Sheng Li. Self-rewarding language models. *arXiv preprint arXiv:2502.06754*, 2025.

[15] Arjun Prasad, Deepak Kumar, Lei Wang, and Fangzhou Tang. Self-improving language models via preference optimization. *arXiv preprint arXiv:2412.07890*, 2024.

[16] Ming Chen, Qi Luo, Hao Zhang, Yiming Yang, and Zhen Xie. Self-play fine-tuning for instruction-following language models. *arXiv preprint arXiv:2411.05678*, 2024.

[17] Rafael Rafailov, Archit Sharma, Eric Mitchell, and Stefano Ermon. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[18] Dan Xu, Yifan Li, Jun Chen, Wei Zhao, and Xiaodong Liu. Genius: Self-supervised reasoning with reinforcement-like methods. *arXiv preprint arXiv:2503.09876*, 2025.

[19] Ting Zhang, Min Huang, Kai Zhang, Chen Wu, and Hong Li. The right for the right reason: Semi-supervised reasoning in llms. *arXiv preprint arXiv:2504.01234*, 2025.

[20] Ling Zhao, Zhiqi Sun, Wei Chen, Ping Zhou, and Xinyu Gu. Absolute zero: Reinforced self-play for reasoning models. *arXiv preprint arXiv:2505.02345*, 2025.

[21] Yiming Wang, Tao He, Qian Lin, Jie Xu, and Jun Gao. Reinforcement learning with minimal supervision for mathematical reasoning. *arXiv preprint arXiv:2503.04567*, 2025.

A Related Work

Reinforcement Learning [5] has become central to improving the reasoning and instruction-following abilities of large language models. The most prominent example, Reinforcement Learning from Human Feedback (RLHF) [6], aligns models with human preferences through reward modeling and optimization methods such as Proximal Policy Optimization (PPO) [7]. More recently, Large Reasoning Models (LRMs) like DeepSeek-R1 [2] have highlighted the role of RL in strengthening multi-step reasoning, often by leveraging rule-based rewards. A notable case is GRPO [1], which, unlike RLHF's broad focus on open-domain alignment, explicitly targets mathematical problem solving by encouraging long chain-of-thought (CoT) [8] generation. Considerable attention has since been devoted to improving the robustness and stability of such rule-based RL methods, including GRPO and PPO [9, 10, 11]. Despite these advances, training still relies on supervised data, while inference requires models to extrapolate extended reasoning on unseen tasks. Moreover, effective reward design remains limited to domains with verifiable outcomes—such as math or code—where correctness can be automatically checked [12, 13].

Beyond verifiable tasks, researchers have investigated self-rewarding [14, 15] and self-play [16] approaches for unlabeled data. However, most of these efforts either emphasize general instruction following [14, 16] or employ preference-based optimization strategies like DPO [17] instead of online RL. In parallel, several concurrent directions [18, 19, 20] explore semi-supervised or reinforcement-inspired reasoning. TTRL [3] is one of the latest work in this direction, which use majority voting as a self-derived reward signal, which reduces susceptibility to reward hacking. Complementary findings by [21] show that even a single reward-providing example can substantially improve mathematical reasoning, underscoring the potential of lightweight supervision. Intuitor [4] replaces external rewards with the model's intrinsic self-certainty—a measure of confidence defined as the average KL divergence from a uniform distribution over the vocabulary. In this work, we build upon the concept of majority voting to generate pseudo-labels, similar to TTRL [3], and propose a new framework to address the limitations arising from noisy pseudo-labels.

B Data Generation Pipeline

To support our curriculum-guided reinforcement learning approach, we developed a data curation pipeline to generate synthetic unlabeled datasets with predefined difficulty levels. We leverage an LLM to create diverse prompts, using a structured prompting strategy that emphasizes generating high-quality, varied samples. The prompt explicitly instructs the LLM to produce prompts aligned with a specified difficulty scale (Levels 1 to 5), where each level is defined by example problems provided in the prompt. These examples are carefully selected to represent the reasoning complexity and problem structure characteristic of each difficulty level, ranging from simple arithmetic for Level 1 to advanced multi-step reasoning for Level 5. To ensure robust dataset creation, we generate batches of 25 samples per iteration, allowing for sufficient volume while maintaining computational efficiency.

To mitigate bias toward the provided example prompts and promote diversity in the generated dataset, we dynamically refresh the example problems included in each prompting iteration. This dynamic sampling approach ensures that the LLM does not overfit to specific patterns in the provided examples, resulting in a broader range of problem types and structures within each difficulty bin. The generated prompts are then partitioned into $K = 5$ difficulty bins, $\mathcal{D}_1, \ldots, \mathcal{D}_K$, based on their alignment with the defined difficulty criteria. This curated dataset is used to train models in our **CuMa** framework, enabling a progressive learning curriculum that aligns with the model's reasoning capacity and enhances generalization across tasks.

Setting	Performance
Ours	32.8
w/o reward masking	30.7
w/o curated data	24.5
w/o curriculum	20.1

Table 2: Ablation study

The prompt used for curating our dataset is provided below:

```
Data curation prompt:

"""
You are a math reasoning question generator for LLM training.  Generate few
high-quality reasoning questions that should be self-contained, promote
step-by-step thinking, and not require external knowledge beyond basic facts.
Here are the texts:

Key requirements:
- Do not provide answers, solutions, or reasoning chains.  Output only the
questions with their difficulty labels.
- Vary the questions to cover different sub-topics, including but not limited to
('Algebra', 'Counting  Probability', 'Geometry', 'Intermediate Algebra', 'Number
Theory', 'Prealgebra', 'Precalculus')
- Ensure questions are original and engaging.
- Include a difficulty level for each question on a scale of 1-5 (1:  very easy,
basic logic; 5:  very hard, multi-step or abstract reasoning).
- Target difficulty level:  {target_level}.
- Examples of level 1 questions:
{level_1_examples}
- Examples of level 2 questions:
{level_2_examples}
- Examples of level 3 questions:
{level_3_examples}
- Examples of level 4 questions:
{level_4_examples}
- Examples of level 5 questions:
{level_5_examples}

The examples above are for illustration only, and to distinguish between
different difficulty levels.  Your generated questions must be different from
these examples.

Output format:
- Level {target_level}; Type:  [Sub-topic]; [Question text]
- Level {target_level}; Type:  [Sub-topic]; [Question text]
...  (repeat for N questions)

Generate exactly N questions of Level {target_level}.  """
```

C Experiments

C.1 Implementation details

To implement our **CuMa** method, we apply Group Relative Policy Optimization independently across each benchmark, adapting the approach outlined in [2]. We utilize a cosine learning rate schedule with a peak learning rate of 3×10^{-6} and employ the AdamW optimizer for policy optimization. For each training prompt, we generate 8 candidate responses using a temperature of 0.6 for majority voting to estimate pseudo-labels. The maximum generation length is capped at 3,072 tokens for all models. The number of training episodes is set to 1. All experiments were conducted on a cluster of 4 NVIDIA H100 80GB GPUs.

C.2 Ablation study

To evaluate the contributions of each component in our **CuMa** framework, we conducted an ablation study on the Math 500 benchmark using the Qwen2.5-0.5B model, with results summarized in Table 2. Our full method achieves a performance score of 32.8. Removing the reward-masking mechanism, which excludes samples without a majority consensus, reduces the score to 30.7, indicating that

masking inconclusive rollouts is crucial for stabilizing training and preventing learning from noisy feedback. Omitting the curated synthetic data generated by our data curation pipeline further degrades performance to 24.5, highlighting the importance of diverse, difficulty-controlled samples in enhancing reasoning capabilities. Finally, excluding the curriculum learning strategy, which progressively introduces harder problems, results in a significant drop to 20.1, underscoring the necessity of aligning training data difficulty with the model's capacity to avoid performance degradation. These results confirm that each component—reward masking, curated data, and curriculum learning—plays a critical role in the effectiveness of **CuMa** for label-free reinforcement learning across varying model sizes.