Eliciting Reasoning in LLMs using Logprob-based Rewards

Jérémy Barghorn Theo Schifferli Lindsay Bordier Aymeric de Chillaz
Zeineb Mellouli

EPFL

Introduction

Despite increasing model scale and the use of CoT prompting [1, 2, 3], LLMs still struggle with consistent reasoning. We build on DeepSeek-R1 and Group Relative Policy Optimization (GRPO) [4, 5], applying logprob-based rewards to train models based on internal confidence rather than task-specific rules. We use the **Qwen3-1.7B** model for all experiments, selected for its strong reasoning potential and reliable adherence to structured output formats. Our method generalizes across domains—structured symbolic math and open-ended poetry—without handcrafted evaluators.

Reward Mechanisms

Structured Output Format. Each model response is formatted as:

```
<think>Reasoning </think><answer>Final answer </answer>
```

Format Reward. All training runs include a binary Format Reward that checks for correct use of `<think>` and `<answer>` tags.

Rule-Based Rewards. Task-specific evaluators provide interpretable and targeted feedback:

- **Math:** Evaluates use of input numbers and exact correctness of the final result.
- **Poetry:** Combines rhyme, syllable count, form classification, and semantic similarity.

Logprob-Based Rewards. Used as a domain-agnostic signal based on model confidence. For prompt p, reasoning r, and gold answer a:

$$R = \log P(a \mid p, r), b = \frac{1}{N}\sum_{i=1}^{N} \log P(a_i \mid p, r_i), A = R - b$$

This reward encourages reasoning traces that make a more likely, independent of the model's own answer.

Logprob Reward Normalization

Logprobs are negative and scale with sequence length. This introduces bias, especially in tasks like poetry where gold answer lengths vary.

- **Length Normalization:** Average the total logprob over the number of tokens in the gold answer.
- **Exponentiation:** Convert normalized logprobs to $[0, 1]$ reward scale via $\exp(\cdot)$.

Batch-Level Normalization: Using batch size 64, we apply:

- **Z-Score:** Mean-center and scale to highlight outliers early in training.
- **Min-Max:** Rescale to $[0, 1]$ to emphasize strong vs. weak generations throughout training.

Finding: Min-max + length normalization yields the most stable reward progression across training.

Failure Case: Without length normalization, the model exploited reward structure by producing repeated `<think>` blocks without valid answers—optimizing logprob while ignoring format. We constrained generations to one `<think></think>` block to mitigate this.

Custom GRPOTrainer: Reasoning-Aware Masking

The default GRPO loss aggregates token-level advantages over the entire output sequence. However, in our logprob-based reward setup, the reward is based only on the reasoning trace—not the generated answer. Including tokens beyond the `</think>` tag introduces noisy gradients, especially when incorrect answers are predicted confidently.

Key Modification. We implement a `CustomGRPOTrainer` that masks out all tokens following the final reasoning step. Only tokens up to the end of the reasoning trace are used in the loss computation.

Modified GRPO Objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G} \mathcal{L}^{(i)}_{\text{GRPO}}(\theta)$$

$$\mathcal{L}^{(i)}_{\text{GRPO}}(\theta) = \frac{1}{|o_i^{\leq t_{\text{end}}}|}\sum_{t=1}^{t_{\text{end}}} \ell_{i,t} - \beta D_{\text{KL}}[\pi_\theta \parallel \pi_{\text{ref}}]$$

$$\ell_{i,t} = \min\left[\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \cdot \hat{A}_{i,t},\ g(\epsilon, \hat{A}_{i,t})\right]$$

Explanation:

- $o_i^{\leq t_{\text{end}}}$: output truncated before the end of the reasoning (`</think>`).
- $\hat{A}_{i,t}$: token-level advantage.
- $g(\epsilon, \cdot)$: PPO-style clipping function.
- $D_{\text{KL}}[\pi_\theta \parallel \pi_{\text{ref}}]$: KL penalty to stabilize updates.

This design ensures alignment between training signals and logprob-based rewards by focusing optimization on the reasoning trace only.

Task Setup

We evaluate models on two distinct reasoning tasks:

Math Task (Structured Reasoning): Given 4 numbers and a target, the model must produce a valid arithmetic expression using each number exactly once to reach the target. Outputs are evaluated for correctness and reasoning trace structure.

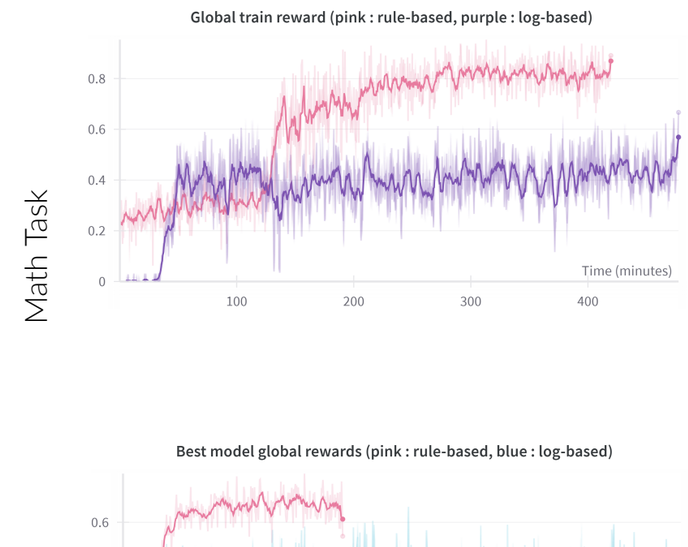
Poetry Task (Creative Reasoning): Given the beginning of a poem (with author, title, and form), the model must generate a stylistically consistent and semantically coherent ending. Outputs are evaluated using rhyme, syllable count, form adherence, and embedding similarity to the gold ending.

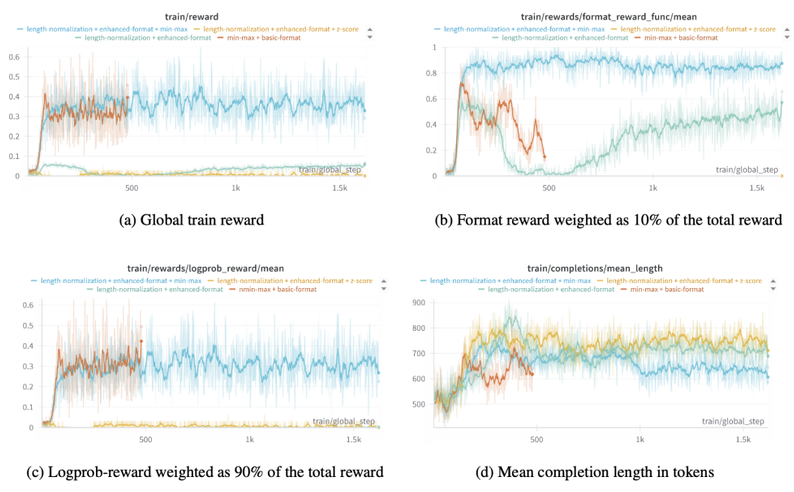
Results & Evaluation Overview

We compare rule-based and logprob-based GRPO across math and poetry tasks. While both reward schemes improve over the baseline, rule-based GRPO consistently achieves the best scores—even in poetry—thanks to its strict format alignment and targeted feedback.

Logprob-based GRPO, though slightly behind in benchmarks, showed robust learning dynamics and produced stylistically coherent outputs. Its reward structure—based on model confidence rather than handcrafted rules—encourages flexible reasoning patterns.

We believe logprob-based rewards are especially promising for **more creative tasks**, such as storytelling, dialogue generation, or speculative writing, where rigid correctness is ill-defined. In such contexts, diversity and coherence matter more than exact match, making logprob-guided learning a more natural fit.





(a) Global train reward

(b) Format reward weighted as 10% of the total reward

(c) Logprob-reward weighted as 90% of the total reward

(d) Mean completion length in tokens

Benchmark Evaluation Scores

Model	Math Score	Poetry Score
Baseline	0.009	0.000
Rule-based GRPO	0.449	0.091
Logprob GRPO	0.301	0.046

References

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[2] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[3] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

[4] DeepSeek-AI et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[5] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.