# LEARNING BEFORE REWARD: UNREWARDED EXPLORATION IN LARGE LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Recent advances in the reasoning capabilities of large language models (LLMs) have been driven largely by reinforcement learning with verifiable rewards (RLVR). However, it remains unclear whether LLMs can improve through exploration even in the absence of rewards (i.e., neither external verifier rewards nor internally constructed rewards), and whether such exploration can benefit subsequent optimization once rewards are introduced. Blodgett's classic maze experiments showed that rats could acquire knowledge of their environment even before receiving rewards. Motivated by this finding, we study a training setting in which LLMs first undergo training without rewards, called *unrewarded exploration*, where LLMs learn from their own generated responses without any reward signals, and are then further optimized once rewards are introduced. Across several model families and task domains, we find that LLMs achieve measurable performance gains from the unrewarded exploration phase, and that these gains translate into further improvement once rewards are introduced. In addition, models trained under this setting ultimately outperform counterparts trained with standard RL throughout the training process. To explain these empirical findings, we provide theoretical analyses showing why unrewarded exploration can yield performance gains, shedding light on these dynamics. Taken together, our results suggest that LLMs may benefit from an initial phase of unrewarded exploration, with performance gains becoming more pronounced when rewards are introduced.

## 1 INTRODUCTION

Recent advances in large language models (LLMs) post-training have been driven by reinforcement learning (Bai et al., 2022; Achiam et al., 2023; Guo et al., 2025; Yang et al., 2025; Team et al., 2025), especially in domains such as mathematical reasoning and agentic decision making. In these settings, performance improvements are typically achieved by optimizing models against reward signals (Schulman et al., 2017; Shao et al., 2024), such as outcome rewards and task-specific verifiers. Despite the empirical success of this paradigm, it remains unclear whether reward signals are strictly necessary for improving model performance during post-training. In particular, an important open question is whether LLMs can benefit from an initial phase of training without rewards, and whether such a phase can improve subsequent training once rewards are introduced.

This question is inspired by Blodgett's classic rat experiment (Blodgett, 1929), in which rats exploring a maze without rewards (i.e., food) showed little reduction in error (measured by entries into dead ends), and later the error curve dropped sharply once reward was introduced. This study draws our inspiration from this broad phenomenon at the level of training dynamics, without assuming that LLMs and rats share the same underlying learning mechanism. We ask an analogous question for LLMs training: *whether an initial phase of training without rewards (i.e., neither external verifier rewards nor internally constructed rewards) produce performance gains, and can subsequent training with rewards introduced after such exploration ultimately yield better performance than training with rewards throughout?* We begin by studying this question empirically in a setting where LLMs are initially trained without rewards and are then further trained once rewards are introduced.

We refer to the initial phase as *unrewarded exploration*, during which LLMs learn from their own sampled responses without reward assignment. Since no rewards are assigned to reward or penalize the generated responses, LLMs are allowed to explore diverse behaviors. During this phase, unrewarded



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

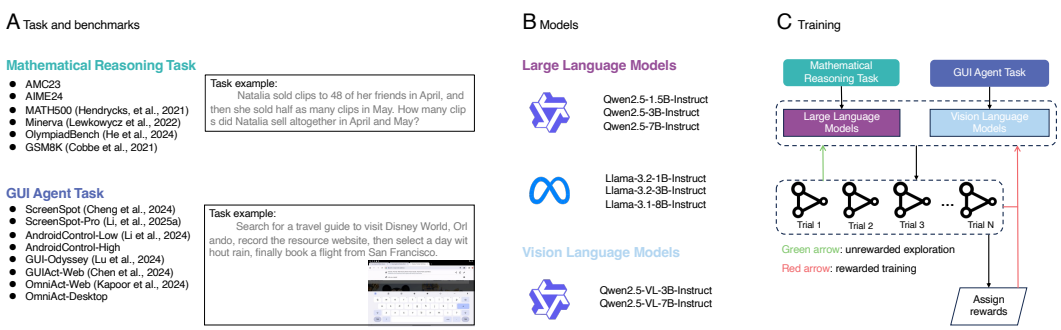


Figure 1: Overview of tasks and models we use in this work, and training paradigm. **A**: Examples for the different tasks. Experimental tasks include mathematical reasoning and GUI agent task. **B**: The LLMs used include different series and parameter sizes. **C**: Our training process. For every task, questions are given to LLMs to generate a group of responses. The green arrow represents training without rewards. The red arrow represents training with rewards.

exploration is implemented as a GRPO-style (Shao et al., 2024) update in which all sampled responses are treated by fixing the advantage to a constant, rather than assigning rewards or filtering responses. We evaluate this setting across several model families and representative task domains, including mathematical reasoning and GUI (Graphical User Interface) agent tasks. The first is mathematical reasoning, a domain widely used for RL in LLMs, where performance can be measured cleanly through problem-solving accuracy in a symbolic setting (Hendrycks et al., 2021; Cobbe et al., 2021; He et al., 2024; Lewkowycz et al., 2022). The second is GUI agent task, where models must act like humans to manipulate computers and mobile devices, extending our experiments to interactive environments (Cheng et al., 2024; Wu et al., 2025; Li et al., 2025; 2024; Lu et al., 2025; Chen et al., 2025; Kapoor et al., 2024). These tasks span both low-level single-step grounding and high-level multi-step decision making: in low-level tasks, LLMs are required to perform a single correct action (i.e., to ground the area in which to perform a click), while in high-level tasks, LLMs must execute a sequence of actions (e.g., click, type, close, etc.), adapting iteratively to changing environments until the goal is achieved. Experiments show that LLMs can achieve measurable and surprising performance gains from the initial unrewarded exploration phase, and that these gains become more pronounced once rewards are introduced. Moreover, in our controlled comparisons, introducing rewards after unrewarded exploration yields better performance than training with rewards throughout.

Beyond these empirical findings, we provide theoretical analyses that explain why unrewarded exploration can improve performance. Specifically, we show that, under monotonicity conditions relating the proposal distribution to latent utility, constant-advantage GRPO updates with clipping and KL regularization can improve expected latent utility even without rewards. These analyses offer a principled account of the dynamics observed in our experiments across several model families and task domains. Overall, our study highlights unrewarded exploration as a striking and previously underexplored phenomenon in LLMs post-training, motivated by Blodgett's rat experiments (Blodgett, 1929) but studied here in a modern machine learning setting.

In summary, our contributions are as follows.

- We first formulate and study *unrewarded exploration* as a GRPO-style post-training phase in which LLMs learn from self-generated responses without reward assignment.
- We present empirical evidence that LLMs can achieve performance gains during an initial phase of *unrewarded exploration*. We further show that introducing rewards after unrewarded exploration can ultimately yield better performance than training with rewards throughout.
- We provide theoretical analyses that explain why unrewarded exploration can improve performance, offering theoretical insight into the observed dynamics.
- We conduct extensive experiments showing that unrewarded exploration generalizes across model families, model parameter sizes, and task domains.

# 2 RELATED WORK

## 2.1 REINFORCEMENT LEARNING FOR LLMS

Reinforcement learning from human feedback (RLHF) has emerged as a central method for aligning and fine-tuning LLMs. Early work by Ziegler et al. (2019) demonstrated that human preference data combined with reinforcement learning substantially improved model alignment performance. Most RLHF implementations rely on Proximal Policy Optimization (PPO) (Schulman et al., 2017), which has become the universally acknowledged standard for policy optimization in many domains. More recent studies have proposed improvements and alternatives to PPO, such as GRPO (Shao et al., 2024), which are attributed to the powerful capabilities of the DeepSeek-R1 model and the phenomenon of self-reflection (Guo et al., 2025), followed by a series of works (Yu et al., 2025; Xiong et al., 2026; Zheng et al., 2025). These methods have been applied to domains such as mathematical reasoning, code generation and other agent tasks (Luo et al., 2025; Zeng et al., 2025; Zhang et al., 2026a), but they all presuppose that reward signals are indispensable for driving performance improvement.

## 2.2 RL FOR LLMS BEYOND CONVENTION

Recent work has begun to explore whether LLMs can be optimized while reducing dependence on conventional external rewards. Agarwal et al. (2025) show that entropy minimization alone can improve reasoning performance without labeled data, through RL with negative entropy as the reward, suggesting that reasoning capabilities can be elicited from pretrained models using purely internal signals. Zhao et al. (2026) also propose RL from internal feedback, in which LLMs' own self-certainty serves as the reward signal, demonstrating that reasoning can be improved without external rewards or labeled data. Absolute Zero Reasoner (AZR) (Zhao et al., 2025) further weakens reliance on human-labeled data by allowing LLMs to generate their own tasks and learn from them in a self-play manner, while using a code executor to validate tasks and verify answers as a unified source of verifiable reward. These studies are closely related to ours in that they all move beyond standard externally annotated reward pipelines. Unlike prior methods that replace task rewards with alternative signals, our unrewarded exploration introduces no behavior-dependent intrinsic reward, novelty score, or coverage objective, and does not use any response-dependent evaluation. All responses receive the same constant advantage and are not differentiated by correctness, surprise, novelty, or informativeness. Please refer to Appendix E.2 for more discussion.

# 3 METHOD

## 3.1 REWARDED TRAINING

When training LLMs with reward assignment, we adopt the training objective proposed by Shao et al. (2024) as our reinforcement learning objective with rewards:

$$\mathcal{J}_{\text{Rewarded}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\ \{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\},$$

where $q$, $a$, $\mathcal{D}$, $o$, $G$, $\pi_{\text{old}}$, $\pi_{ref}$ and $\pi_\theta$ represent question, answer, dataset, model response, group size, old policy model, reference model, and current policy model, respectively. The clip operation is designed to constrain the input variable within the predefined closed interval $[1-\varepsilon, 1+\varepsilon]$, and $\varepsilon$ is a hyper-parameter introduced in PPO (Schulman et al., 2017) for stabilizing training. $r_{i,t}(\theta)$ is computed as $r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\text{old}}(o_{i,t}|q,o_{i,<t})}$. The advantage $\hat{A}_{i,t}$ is computed as $\hat{A}_{i,t} = \frac{r_i - \text{mean}\{R\}_{i=1}^{G}}{\text{std}\{R\}_{i=1}^{G}}$, where $r_i$ is the reward assigned by the reward rules for the response $o_i$. And the KL-divergence $\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]$ is computed by

$$\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta(o_{i,t} \mid q, o_{i,<t})} - 1.$$

## 3.2 UNREWARDED EXPLORATION

In RL training, rewards are used to calculate the advantage value for each response (as expressed in Section 3.1). When no rewards are assigned for all responses, each response should be treated

equally. During this stage, we adopt the same objective as in rewarded training but set the advantage value to 1. That means our training objective during unrewarded exploration is

$$\mathcal{J}_{\text{Unrewarded}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\ \{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[r_{i,t}(\theta), \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\},$$

where all symbols in the training objective remain the same as those in Section 3.1.

# 4 Experiments

## 4.1 Models for Experiments

The models we use in all experiments include the Qwen series and the LLaMA series, both of which are open-source models and can be downloaded from the Hugging Face website (https://huggingface.co). The Qwen series includes Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct while the LLaMA series includes Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct. These choices span multiple model families and a broad range of model scales, allowing us to examine whether the effects of unrewarded exploration generalize across model series and parameter sizes. All experiments are initialized from publicly available models, and we do not perform any additional SFT before unrewarded exploration or rewarded training.

Table 1: Model pass@1 performance across various mathematical reasoning benchmarks, reported from the single checkpoint that achieves the highest average score.

| Model | GSM8K | MATH500 | Minerva | OlympiadBench | AIME24 | AMC23 | Avg |
|---|---|---|---|---|---|---|---|
| *Llama 1B Models* | | | | | | | |
| Llama-3.2-1B-Instruct | 26.8 | 11.8 | 1.8 | 3.7 | 0.0 | 2.5 | 7.8 |
| + Unrewarded | 33.0 (+6.2) | 15.0 (+3.2) | 5.9 (+4.1) | 3.6 (-0.1) | 0.0 | 15.0 (+12.5) | 12.1 (+4.3) |
| + Rewarded | 56.1 | 33.2 | 7.0 | 9.2 | 10.0 | 17.5 | 22.2 |
| + Unrewarded → Rewarded | 58.2 (+2.1) | 37.8 (+4.6) | 6.6 (-0.4) | 9.9 (+0.7) | 10.0 | 20.0 (+2.5) | **23.8** (+1.6) |
| *Llama 3B Models* | | | | | | | |
| Llama-3.2-3B-Instruct | 45.6 | 26.2 | 7.7 | 6.4 | 3.3 | 17.5 | 17.8 |
| + Unrewarded | 67.5 (+21.1) | 36.6 (+10.4) | 7.7 | 11.3 (+4.9) | 6.7 (+3.4) | 25.0 (+7.5) | 25.8 (+8.0) |
| + Rewarded | 82.0 | 53.4 | 23.2 | 18.7 | 20.0 | 37.5 | 39.1 |
| + Unrewarded → Rewarded | 83.5 (+1.5) | 51.4 (-2.0) | 16.5 (-6.7) | 16.1 (-2.6) | 26.7 (+6.7) | 42.5 (+5.0) | **39.5** (+0.4) |
| *Llama 8B Models* | | | | | | | |
| Llama-3.1-8B-Instruct | 23.0 | 18.6 | 4.0 | 5.8 | 0.0 | 5.0 | 9.4 |
| + Unrewarded | 64.4 (+41.4) | 26.8 (+10.2) | 11.8 (+7.8) | 5.8 | 0.0 | 25.0 (+20.0) | 22.3 (+12.9) |
| + Rewarded | 88.5 | 55.8 | 29.4 | 19.1 | 13.3 | 37.5 | 40.6 |
| + Unrewarded → Rewarded | 88.5 | 53.8 (-2.0) | 30.5 (+1.1) | 22.5 (+3.2) | 20.0 (+6.7) | 32.5 (-5.0) | **41.3** (+0.7) |
| *Qwen 1.5B Models* | | | | | | | |
| Qwen2.5-1.5B-Instruct | 68.7 | 47.6 | 16.5 | 18.1 | 0.0 | 20 | 28.5 |
| + Unrewarded | 68.2 (-0.5) | 48.0 (+0.4) | 17.3 (+0.8) | 19.3 (+1.2) | 0.0 | 32.5 (+12.5) | 30.9 (+2.4) |
| + Rewarded | 71.0 | 54.4 | 16.5 | 20.1 | 0.0 | 32.5 | 32.4 |
| + Unrewarded → Rewarded | 72.2 (+1.2) | 54.2 (-0.2) | 19.5 (+3.0) | 19.4 (-0.7) | 10.0 (+10.0) | 37.5 (+5.0) | **35.5** (+3.1) |
| *Qwen 3B Models* | | | | | | | |
| Qwen2.5-3B-Instruct | 84.9 | 63.0 | 24.3 | 24.4 | 6.7 | 37.5 | 40.1 |
| + Unrewarded | 85.7 (+0.8) | 61.4 (-1.6) | 25.7 (+1.4) | 27.0 (+2.6) | 6.7 | 50.0 (+12.5) | 42.8 (+2.7) |
| + Rewarded | 84.7 | 65.8 | 25.7 | 28.1 | 13.3 | 45.0 | 43.8 |
| + Unrewarded → Rewarded | 85.6 (+0.9) | 67.6 (+1.8) | 25.7 | 29.3 (+1.2) | 10.0 (-3.3) | 55.0 (+10.0) | **45.5** (+1.7) |
| *Qwen 7B Models* | | | | | | | |
| Qwen2.5-7B-Instruct | 91.6 | 73.6 | 37.9 | 36.7 | 6.7 | 47.5 | 49.0 |
| + Unrewarded | 91.0 (-0.6) | 77.4 (+3.8) | 37.1 (-0.8) | 36.4 (-0.3) | 20.0 (+13.3) | 50.0 (+2.5) | 52.0 (+3.0) |
| + Rewarded | 91.7 | 76.6 | 36.8 | 43.1 | 23.3 | 57.5 | 54.8 |
| + Unrewarded → Rewarded | 91.7 | 77.6 (+1.0) | 44.1 (+7.3) | 40.1 (-3.0) | 20.0 (-3.3) | 60.0 (+2.5) | **55.6** (+0.8) |

## 4.2 Datasets for Experiments

### 4.2.1 Mathematical Reasoning

We use the training data from SimpleRL-Zoo-Data (Zeng et al., 2025) collection for training, which was constructed from the training data of GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). For the Qwen series models, we use the simplelr_qwen_level3to5 dataset, and for the LLaMA series models, we use the simplelr_abel_level3to5 dataset. However, the mathematical reasoning problems and ground truth in both training datasets are the same. More details about the two datasets can be found in Appendix F. We evaluate the trained models on AMC23, AIME24, GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024) and Minerva Math (Lewkowycz et al., 2022) benchmarks.

### 4.2.2 GUI Agent Task

We use the dataset from Luo et al. (2025) for training Vision Language Models, which was collected and filtered from FindWeb (Penedo et al., 2024), UIBert (Rastogi et al., 2021), AMEX (Chai et al., 2025), RICOSCA (Li et al., 2020), Seeclick (Cheng et al., 2024) and OS-Otlas (Wu et al., 2025). We evaluate the trained models on ScreenSpot (Cheng et al., 2024), ScreenSpot-Pro (Li et al., 2025), AndroidControl-Low (Li et al., 2024), AndroidControl-High, GUIAct-Web (Li et al., 2025), OmniAct-Web, OmniAct-Desktop (Kapoor et al., 2024) and GUI-Odyssey (Lu et al., 2025) benchmarks, containing simple grounding tasks, low-level tasks and high-level tasks.

Table 2: Model pass@1 performance across various mathematical reasoning benchmarks, reported by the best accuracy achieved on each benchmark over the training process.

| Model | GSM8K | MATH500 | Minerva | OlympiadBench | AIME24 | AMC23 | Avg |
|---|---|---|---|---|---|---|---|
| *Llama 1B Models* | | | | | | | |
| Llama-3.2-1B-Instruct | 26.8 | 11.8 | 1.8 | 3.7 | 0.0 | 2.5 | 7.8 |
| + Unrewarded | 33.1 (+4.3) | 18.6 (+6.8) | 6.2 (+4.4) | 5.5 (+1.8) | 3.3 (+3.3) | 15.0 (+12.5) | 13.6 (+5.8) |
| + Rewarded | 57.0 | 37.0 | 9.2 | **11.7** | 10.0 | **25.0** | 25.0 |
| + Unrewarded → Rewarded | **59.4** (+2.4) | **38.8** (+1.8) | **10.3** (+1.1) | 11.6 (-0.1) | **13.3** (+3.3) | **25.0** | **26.4** (+1.4) |
| *Llama 3B Models* | | | | | | | |
| Llama-3.2-3B-Instruct | 45.6 | 26.2 | 7.7 | 6.4 | 3.3 | 17.5 | 17.8 |
| + Unrewarded | 67.5 (+21.9) | 40.8 (+14.6) | 11.0 (+3.3) | 12.1 (+5.7) | 10.0 (+6.7) | 25.0 (+7.5) | 27.7 (+9.9) |
| + Rewarded | **84.2** | 53.8 | **23.2** | 19.7 | 23.3 | 40.0 | 40.7 |
| + Unrewarded → Rewarded | 83.7 (-0.5) | **55.2** (+1.4) | 21.3 (-1.9) | **19.9** (+0.2) | **26.7** (+3.4) | **42.5** (+2.5) | **41.6** (+0.9) |
| *Llama 8B Models* | | | | | | | |
| Llama-3.1-8B-Instruct | 23.0 | 18.6 | 4.0 | 5.8 | 0.0 | 5.0 | 9.4 |
| + Unrewarded | 64.5 (+41.5) | 31.2 (+12.6) | 12.9 (+8.9) | 8.6 (+2.8) | 6.7 (+6.7) | 25.0 (+20.0) | 24.8 (+15.4) |
| + Rewarded | **90.1** | 59.2 | 31.2 | 22.5 | **20.0** | **40.0** | 43.8 |
| + Unrewarded → Rewarded | 89.6 (-0.5) | **59.8** (+0.6) | **33.1** (+1.9) | **23.1** (+0.6) | **20.0** | **40.0** | **44.3** (+0.5) |
| *Qwen 1.5B Models* | | | | | | | |
| Qwen2.5-1.5B-Instruct | 68.7 | 47.6 | 16.5 | 18.1 | 0.0 | 20.0 | 28.5 |
| + Unrewarded | 70.1 (+1.4) | 51.6 (+4.0) | 17.3 (+0.8) | 19.3 (+1.2) | 0.0 | 32.5 (+12.5) | 31.8 (+3.3) |
| + Rewarded | 73.5 | 54.8 | 19.1 | 20.3 | 3.3 | 32.5 | 33.9 |
| + Unrewarded → Rewarded | **75.3** (+1.8) | **58.4** (+3.6) | **19.5** (+0.4) | **22.1** (+1.8) | **10.0** (+6.7) | **40.0** (+7.5) | **37.6** (+3.7) |
| *Qwen 3B Models* | | | | | | | |
| Qwen2.5-3B-Instruct | 84.9 | 63.0 | 24.3 | 24.4 | 6.7 | 37.5 | 40.1 |
| + Unrewarded | 85.7 (+0.8) | 66.8 (+3.8) | 26.8 (+2.5) | 28.3 (+3.9) | 6.7 | 50.0 (+12.5) | 44.1 (+4.0) |
| + Rewarded | 86.6 | 68.4 | 28.7 | 31.9 | **16.7** | 50.0 | 47.1 |
| + Unrewarded → Rewarded | **87.6** (+1.0) | **70.2** (+1.8) | **31.2** (+2.5) | **34.8** (+2.9) | 13.3 (-3.4) | **55.0** (+5.0) | **48.7** (+1.6) |
| *Qwen 7B Models* | | | | | | | |
| Qwen2.5-7B-Instruct | 91.6 | 73.6 | 37.9 | 36.7 | 6.7 | 47.5 | 49.0 |
| + Unrewarded | 92.1 (+0.5) | 77.8 (+4.2) | 42.3 (+4.4) | 38.8 (+2.1) | 20.0 (+13.3) | 57.5 (+10.0) | 54.8 (+5.8) |
| + Rewarded | **93.3** | **79.6** | 38.2 | **43.1** | **23.3** | 62.5 | 56.7 |
| + Unrewarded → Rewarded | 92.9 (-0.4) | 79.2 (-0.4) | **44.1** (+5.9) | 41.6 (-1.5) | **23.3** | **67.5** (+5.0) | **58.1** (+1.4) |

### 4.3 Reward Rules

#### 4.3.1 Mathematical Reasoning Task

The reward rule for training LLMs on mathematical reasoning task is solely based on accuracy: the reward is assigned only if the response answer is correct, following Zeng et al. (2025).

#### 4.3.2 GUI Agent Task

We adopt the reward rules from Luo et al. (2025), which means that the reward is assigned by $r = \alpha r_{format} + \beta r_{acc}$, where $\alpha = 0.2$ and $\beta = 0.8$. $r_{format}$ is assigned 1 if the response format follows the requirements (i.e., thinking process within <think></think> tags and final answer within <answer></answer> tags), and $r_{acc} = r_{act} + r_{point} + r_{text}$. $r_{act}$ is assigned 1 if the predicted action is the same as the ground truth action. $r_{point}$ is assigned 1 if the predicted point is in the ground-truth bounding box (i.e., the predicted [x, y] in ground truth [x1, y1, x2, y2]). $r_{text}$ is assigned 1 if the $F_1$ score between the predicted text and the ground truth text is larger than 0.5.

### 4.4 Hyper-parameters and Computational Environment

We provide detailed parameter settings here. All training in our experiments uses the AdamW optimizer with weight_decay $= 0.01$, and betas $= (0.9, 0.999)$. For mathematical reasoning task, we train models with the same parameters for both unrewarded exploration and rewarded training. Key hyper-parameters include learning_rate $= 5 \times 10^{-7}$, KL coefficient $\beta = 0.001$, batch_size $= 256$, max_response_length $= 3072$, group_size $= 4$ and generation_temperature $= 1.0$, and detailed training budget can be found in Appendix C. For GUI agent task, we also train models with the same parameters for both unrewarded exploration and rewarded training. Key hyper-parameters include steps $= 120$ (i.e., Unrewarded + Rewarded = 120, Rewarded = 120, respectively), learning_rate $= 1 \times 10^{-6}$, KL coefficient $\beta = 0.01$, batch_size $= 128$, max_response_length $= 1024$, group_size $= 5$, generation_temperature $= 1.0$. All experiments are conducted on 8 NVIDIA-A800-80G GPUs.

### 4.5 Results

We use the following notation in all result tables. Rewarded: training with rewards throughout. Unrewarded: training without rewards, neither external verifier rewards nor internally constructed rewards (i.e., *unrewarded exploration*). We show the performance gains in the + *Unrewarded* row compared to the base and the + *Unrewarded* → *Rewarded* row compared to the + *Rewarded* row. We provide our main experimental results to answer our research question in this section. Extended experiments can be found in Appendix B.

#### 4.5.1 Mathematical Reasoning Task

Table 1 reports results from the single checkpoint that achieves the highest average performance across all six mathematical reasoning benchmarks, comparing the resulting models. Table 2 reports the best accuracy achieved on each benchmark over the entire training process, comparing the method itself. It can be seen that in the mathematical reasoning task, the performance of LLMs on the evaluation benchmarks is improved to some extent even after LLMs undergo unrewarded exploration, which corresponds to the experiments of rats in the absence of food rewards in Blodgett's experiments (Blodgett, 1929). That is to say, LLMs can improve their performance simply by learning

Table 3: Results on Screenspot-Pro.

| Model | Development | | Creative | | CAD | | Scientific | | Office | | OS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Text | Icon | Text | Icon | Text | Icon | Text | Icon | Text | Icon | Text | Icon |
| *Qwen 3B Models* | | | | | | | | | | | | |
| Qwen2.5-VL-3B-Instruct | 14.3 | 0.7 | 23.7 | 2.1 | 11.2 | 3.1 | 34.7 | 8.2 | 21.5 | 3.8 | 14.0 | 2.3 |
| + Unrewarded | 22.1 (+7.8) | 2.8 (+2.1) | 30.3 (+6.6) | 4.9 (+2.8) | 15.2 (+4.0) | 7.8 (+4.7) | 46.5 (+11.8) | 11.8 (+3.6) | 33.3 (+11.8) | 11.3 (+7.5) | 19.6 (+5.6) | 6.7 (+4.4) |
| + Rewarded | 33.8 | 4.8 | 41.4 | 2.8 | 26.9 | 7.8 | **61.8** | **17.3** | **53.7** | 17.0 | 27.1 | 6.7 |
| + Unrewarded → Rewarded | **35.1** (+1.3) | **6.2** (+1.4) | **41.9** (+0.5) | **5.6** (+2.8) | **33.0** (+6.1) | **10.9** (+3.1) | 60.4 (-1.4) | **17.3** | 50.9 (-2.8) | **18.9** (+1.9) | **39.0** (+11.9) | **7.9** (+1.2) |
| *Qwen 7B Models* | | | | | | | | | | | | |
| Qwen2.5-VL-7B-Instruct | 30.5 | 2.1 | 22.7 | 3.5 | 10.7 | 3.1 | 39.6 | 6.4 | 36.2 | 11.3 | 29.9 | 5.6 |
| + Unrewarded | 40.9 (+10.4) | 2.8 (+0.7) | 31.3 (+8.6) | 6.3 (+2.8) | 17.8 (+7.1) | 3.1 | 45.8 (+6.2) | 9.1 (+2.7) | 44.6 (+8.4) | 13.2 (+1.9) | 33.7 (+3.8) | 13.5 (+7.9) |
| + Rewarded | **50.7** | **4.8** | 38.9 | **9.1** | 24.9 | **6.3** | 55.6 | 11.8 | 58.8 | **26.4** | **41.1** | **16.9** |
| + Unrewarded → Rewarded | 50.0 (-0.7) | 3.5 (-1.3) | **41.9** (+3.0) | **9.1** | **31.0** (+6.1) | 4.7 (-1.6) | **62.5** (+6.9) | **13.6** (+2.8) | **64.4** (+5.6) | 20.8 (-5.6) | **41.1** | 15.7 (-1.2) |

Table 4: Results on ScreenSpot.

| Model | Mobile | | Web | | Desktop | | Avg |
|---|---|---|---|---|---|---|---|
| | Text | Icon/Widget | Text | Icon/Widget | Text | Icon/Widget | |
| *Qwen 3B Models* | | | | | | | |
| Qwen2.5-VL-3B-Instruct | 73.5 | 50.5 | 71.1 | 31.4 | 88.3 | 58.1 | 62.2 |
| + Unrewarded | 88.7 (+15.2) | 64.6 (+14.1) | 78.3 (+7.2) | 60.2 (+28.8) | 87.1 (-1.2) | 52.1 (-6.0) | 71.8 (+9.6) |
| + Rewarded | **96.3** | **76.4** | 87.4 | 66.0 | **93.8** | 57.1 | 79.5 |
| + Unrewarded → Rewarded | 94.9 (-1.4) | 75.6 (-0.8) | **87.8** (+0.4) | **68.5** (+2.5) | 92.3 (-1.5) | **59.3** (+2.2) | **79.7** (+0.2) |
| *Qwen 7B Models* | | | | | | | |
| Qwen2.5-VL-7B-Instruct | 93.8 | 70.3 | 87.0 | 68.0 | 91.2 | 60.0 | 78.4 |
| + Unrewarded | 95.2 (+1.4) | 64.2 (-6.1) | 88.3 (+1.3) | 68.5 (+0.5) | 90.2 (-1.0) | 57.1 (-2.9) | 77.3 (-1.1) |
| + Rewarded | **96.3** | 72.9 | 91.3 | 76.2 | **92.8** | 72.9 | 83.7 |
| + Unrewarded → Rewarded | **96.3** | **79.5** (+6.6) | **91.7** (+0.4) | **78.2** (+2.0) | 92.3 (-0.5) | **75.0** (+2.1) | **85.5** (+1.8) |

Table 5: Results on low-level tasks.

| Model | GUIAct-Web | | | OmniAct-Web | | | OmniAct-Desktop | | | AndroidControl-Low | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Type | GR | SR | Type | GR | SR | Type | GR | SR | Type | GR | SR |
| *Qwen 3B Models* | | | | | | | | | | | | |
| Qwen2.5-VL-3B-Instruct | 56.2 | 65.5 | 56.7 | 51.8 | 49.0 | 48.9 | 55.8 | 49.8 | 49.8 | 61.2 | 74.4 | 59.6 |
| + Unrewarded | 88.2 (+32.0) | 79.4 (+13.9) | 81.3 (+24.6) | 77.4 (+25.6) | 67.0 (+28.0) | 67.0 (+28.0) | 88.0 (+32.2) | 66.4 (+16.6) | 66.4 (+16.6) | 81.2 (+20.0) | **82.8** (+8.4) | **67.4** (+7.8) |
| + Rewarded | **89.5** | 87.2 | 73.9 | **87.8** | **74.4** | **74.4** | **91.9** | 74.6 | 74.6 | 82.5 | 81.9 | 62.5 |
| + Unrewarded → Rewarded | 88.2 (-1.3) | **87.6** (+0.4) | **87.1** (+13.2) | 80.6 (-7.2) | 73.7 (-0.7) | 73.4 (-1.0) | 87.6 (-4.3) | **78.8** (+4.2) | **78.8** (+4.2) | **85.5** (+3.0) | 82.6 (+0.7) | 66.9 (+4.4) |
| *Qwen 7B Models* | | | | | | | | | | | | |
| Qwen2.5-VL-7B-Instruct | 86.5 | 84.4 | **84.2** | 80.5 | 70.7 | 70.7 | 81.3 | 78.8 | 78.8 | 83.1 | **87.3** | 72.5 |
| + Unrewarded | 86.0 (-0.5) | 82.4 (-2.0) | 82.3 (-1.9) | 81.4 (+0.9) | 69.6 (-1.1) | 69.6 (-1.1) | 89.2 (+7.9) | 82.4 (+3.6) | 82.4 (+3.6) | 78.6 (-4.5) | 87.0 (-0.3) | **72.7** (+0.2) |
| + Rewarded | **90.8** | **88.3** | 76.1 | **91.3** | 77.0 | 76.6 | **92.4** | 82.5 | 82.5 | **84.9** | 84.0 | 64.0 |
| + Unrewarded → Rewarded | **90.8** | 86.7 (-1.6) | 83.0 (+6.9) | 85.3 (-6.0) | **77.2** (+0.2) | **76.8** (+0.2) | **92.4** | **83.6** (+1.1) | **83.6** (+1.1) | 84.1 (-0.8) | 82.3 (-1.7) | 67.5 (+3.5) |

Table 6: Results on high-level tasks.

| Model | AndroidControl-High | | | GUI-Odyssey | | |
|---|---|---|---|---|---|---|
| | Type | GR | SR | Type | GR | SR |
| *Qwen 3B Models* | | | | | | |
| Qwen2.5-VL-3B-Instruct | 48.8 | 45.9 | 38.6 | 38.3 | 27.2 | 27.3 |
| + Unrewarded | 64.1 (+15.3) | 52.7 (+6.8) | 49.1 (+10.5) | 67.3 (+29.0) | 34.9 (+7.7) | 31.4 (+4.1) |
| + Rewarded | 63.6 | 59.4 | 45.4 | 54.7 | 41.0 | 34.8 |
| + Unrewarded → Rewarded | **69.5** (+5.1) | **60.0** (+0.6) | **50.6** (+6.2) | **68.5** (+13.8) | **45.4** (+4.4) | **41.4** (+6.6) |
| *Qwen 7B Models* | | | | | | |
| Qwen2.5-VL-7B-Instruct | 68.5 | 62.2 | **56.9** | 55.7 | 37.8 | 34.5 |
| + Unrewarded | 67.1 (-1.4) | 59.3 (-2.9) | 56.2 (-0.7) | 57.7 (+2.0) | 37.3 (-0.5) | 33.4 (-1.1) |
| + Rewarded | 71.8 | **65.5** | 50.9 | 65.0 | 43.3 | 38.3 |
| + Unrewarded → Rewarded | **72.4** (+1.4) | 62.7 (-2.8) | 51.0 (+0.1) | **68.1** (+3.1) | **45.9** (+2.6) | **41.4** (+3.1) |

from their own responses, without reward guidance. Further, by introducing rewards after performing unrewarded exploration, LLMs are able to achieve even slightly better performance after training than if they were trained directly with rewards throughout, which is similar to the results of Blodgett's experiments in which rats were introduced to food rewards after exploration in the absence of food. More experimental details about training dynamics comparison and unrewarded exploration across model series and sizes can be found in Appendix C and D.

### 4.5.2 GUI Agent Task

From the experimental results (Table 3 and Table 4 for GUI grounding tasks, Table 5 for GUI low-level tasks and Table 6 for GUI high-level tasks), with all scores in each row coming from the same checkpoint, we can observe that for GUI agent task requiring interaction with the environment and decision-making, LLMs demonstrate improved performance on benchmarks during the unrewarded exploration phase. These improvements are more pronounced than in mathematical reasoning task. Furthermore, experimental results reveal that the smaller model (i.e., 3B) exhibits superior learning outcomes in unrewarded exploration compared to the larger model (i.e., 7B). We have attempted to train the LLaMA series vision language models (e.g. Llama-3.2-11B-Vision-Instruct). However, their built-in safety mechanisms caused them to systematically reject GUI agent instructions, preventing successful training, and therefore experiments on these models are not reported here. The experimental results of LLMs on GUI tasks during the unrewarded exploration phase mirror the rat experiments (Blodgett, 1929): learning occurs even without rewards. Upon introducing rewards, LLMs continue to improve under reward guidance and even slightly outperform models trained exclusively with rewards, which is also analogous to the experiments.

Interestingly, we find that Qwen2.5-VL-3B-Instruct exhibits much larger gains during the unrewarded exploration stage than Qwen2.5-VL-7B-Instruct (e.g., +29.0 vs +2.0 in Table 6). One plausible explanation for this phenomenon is that weak models tend to be more sensitive to optimization signals, because their initial policies are less stable and more easily altered by exploratory updates. In contrast, larger models begin with stronger priors and more coherent policies, which makes unguided exploration induce comparatively smaller behavioral changes. In contrast to the 3B model, Qwen2.5-VL-7B-Instruct performs worse after both unrewarded exploration and reinforcement learning with rewards throughout than the base model on some benchmarks (e.g., AndroidControl-High in Table 6). A plausible explanation is that the Qwen2.5-VL-7B-Instruct model already starts with a relatively strong, well-calibrated policy on these benchmarks, so both the unrewarded exploration and rewarded training can act as noisy or misaligned optimization signals.

## 5 Theoretical Analyses

In this section, we provide theoretical analyses for our experimental settings explaining why LLMs exhibit improved performance during unrewarded exploration for the two domain tasks.

**Definition 1** (Probability Simplex). *For a finite vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$, the probability simplex $\Delta^{|\mathcal{V}|-1}$ is defined as*

$$\Delta^{|\mathcal{V}|-1} = \left\{ \pi : \mathcal{V} \to [0,1] \;\middle|\; \sum_{a \in \mathcal{V}} \pi(a) = 1 \right\}.$$

*In particular, for each token state $s = (x, y_{<t})$, the conditional distribution $\pi(\cdot \mid s)$ is required to belong to $\Delta^{|\mathcal{V}|-1}$:*

$$\pi(\cdot \mid s) \in \Delta^{|\mathcal{V}|-1}, \qquad \sum_{a \in \mathcal{V}} \pi(a \mid s) = 1.$$

For GUI agent task, LLMs are required to respond with an action type from ['complete', 'close/delete', 'press_home', 'click', 'press_back', 'type', 'select', 'scroll', 'enter'], which means that the action type space for response is sparse. Here, we provide a theoretical analysis of the performance gains after unrewarded exploration in a sparse setting in Theorem 1. A detailed proof is provided in Appendix A.

**Theorem 1** (Performance improvement without rewards in sparse setting)**.** *Consider an autoregressive language model policy $\pi_\theta$ that defines, for each input prompt $x$ and token prefix $y_{<t}$, a conditional distribution $\pi_\theta(y_t \mid x, y_{<t})$ over the next token $y_t$. The model is trained with Group Relative Policy Optimization where the per-token advantage is fixed to $A_{i,t} \equiv 1$, ratio clipping is applied with parameter $\varepsilon > 0$, and a KL trust-region penalty of strength $\beta > 0$ is included. Let the reference policy be $\pi_{ref}(y_t \mid x, y_{<t})$, and let the proposal distribution be $\pi_{old}(y_t \mid x, y_{<t})$. Assume that the likelihood ratio $h(x, y_{<t}, y_t) = \frac{\pi_{old}(y_t|x,y_{<t})}{\pi_{ref}(y_t|x,y_{<t})}$ is nondecreasing in the latent utility $u^\star(x, y_{<t}, y_t)$ (see Definition 2 in Appendix A) for every input $x$, every prefix $y_{<t}$, and every token $y_t$.*

*Then the GRPO update produces an updated conditional policy of the closed form*

$$\pi^\star_\theta(y_t \mid x, y_{<t}) \;=\; \min\Big((1+\varepsilon)\,\pi_{old}(y_t \mid x, y_{<t}),\;\; \tau(x, y_{<t})\,\pi_{ref}(y_t \mid x, y_{<t})\Big),$$

*where $\tau(x, y_{<t}) > 0$ is the unique normalizing constant satisfying $\sum_{y_t \in \mathcal{V}} \pi^\star_\theta(y_t \mid x, y_{<t}) = 1$. Moreover, under the monotone-likelihood-ratio condition, the expected performance*

$$J(\theta) \;=\; \mathbb{E}_{x \sim \mathcal{D}}\, \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[ \sum_{t=1}^{|y|} u^\star(x, y_{<t}, y_t) \right]$$

*satisfies $J(\pi^\star_\theta) \;\geq\; J(\pi_{ref})$. Thus even with no scalar reward signal and with constant advantage, ratio-clipped GRPO with KL regularization yields monotone improvement in the expected utility.*

For the mathematical reasoning task, LLMs solve the problems like humans, which means that the response space is the whole vocabulary of LLMs. Hence, we further provide the theoretical analysis on performance gains after unrewarded exploration in a continuous setting in Theorem 2. A detailed proof is provided in Appendix A.

**Theorem 2** (Performance improvement without rewards in continuous setting)**.** *Fix an input distribution $\mathcal{D}$, an autoregressive language model policy $\pi_\theta(y_t \mid x, y_{<t})$ that admits a density with respect to a common $\sigma$-finite base measure on the token space, a reference policy $\pi_{ref}(y_t \mid x, y_{<t})$ with the same property, and a proposal (teacher) policy $\pi_{\text{prop}}(y_t \mid x, y_{<t})$. Train with Group Relative Policy Optimization in which the per-token advantage is fixed to $A \equiv 1$, the ratio is clipped with parameter $\varepsilon > 0$ so that the per-token factor becomes $\min(\rho, 1+\varepsilon)$, and a KL trust-region penalty of strength $\beta > 0$ is included. Assume that for every token state $(x, y_{<t})$ the likelihood ratio $h(x, y_{<t}, y_t) = \pi_{\text{prop}}(y_t \mid x, y_{<t}) / \pi_{ref}(y_t \mid x, y_{<t})$ is comonotone with the latent token utility $u^\star(x, y_{<t}, y_t)$ under the base distribution $\pi_{ref}(\cdot \mid x, y_{<t})$, meaning that $(u^\star(\cdot) - u^\star(\cdot'))(h(\cdot) - h(\cdot')) \geq 0$ almost everywhere. Then the statewise maximizer of the GRPO surrogate has the water-filling form*

$$\pi^\star_\theta(y_t \mid x, y_{<t}) = \min\big((1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}),\; \tau_{x,y_{<t}}\,\pi_{ref}(y_t \mid x, y_{<t})\big),$$
$$\int \pi^\star_\theta(y_t \mid x, y_{<t})\,\mathrm{d}\nu(y_t) = 1,$$

*for a unique normalizing constant $\tau_{x,y_{<t}} > 0$. Moreover, the expected performance*

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}}\, \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \Big[ \sum_{t=1}^{|y|} u^\star(x, y_{<t}, y_t) \Big] = \mathbb{E}_{x, y_{<t}} \int \pi_\theta(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t)\,\mathrm{d}\nu(y_t)$$

*satisfies $J(\pi^\star_\theta) \geq J(\pi_{ref})$. Hence, even without scalar rewards and with constant advantage, ratio-clipped GRPO with KL regularization yields monotone improvement in expected latent utility in continuous output spaces.*

The theoretical analyses above simultaneously examine the feasibility of enhancing LLMs' performance in both discrete spaces (where the action space available to LLMs is predefined in GUI agent task) and continuous spaces (where no constraints are imposed on LLMs' response in mathematical reasoning task) during unrewarded exploration.

## 6 DISCUSSION

Inspired by Blodgett's rat experiments (Blodgett, 1929), we study whether LLMs can benefit from an initial phase of training without rewards and whether subsequent training with rewards introduced after such exploration ultimately yields better performance than training with rewards throughout. Across mathematical reasoning and GUI agent task, our results show that LLMs can indeed achieve performance gains during unrewarded exploration. In many cases, introducing rewards after unrewarded exploration ultimately yields better performance than training with rewards throughout. Together, these findings highlight unrewarded exploration as a meaningful and previously underexplored aspect of LLM post-training dynamics. Our theoretical analyses further provide insights into why such gains can arise during unrewarded exploration.

A primary limitation of our study is that both the empirical experiments and theoretical analysis are mainly grounded in the GRPO (Shao et al., 2024) algorithm. While this choice provides a controlled and interpretable setting for examining latent learning dynamics, it remains unclear whether the same effects would persist under different RL algorithms. Future work could extend our findings to other policy optimization methods, such as PPO (Schulman et al., 2017) or its variants, to test whether unrewarded exploration exhibits similar benefits across distinct training paradigms.

Another limitation concerns the scope of models and tasks examined in this study. Our experiments focus on the Qwen and LLaMA series, with parameter scales up to 8B. While these models provide a useful testbed for controlled comparisons, they may not fully capture how unrewarded exploration behaves in other model families or at larger scales, where stronger pretraining and broader capabilities may lead to different outcomes. Future work could therefore extend our analysis to a wider range of model series and larger model sizes to examine how these training dynamics evolve with scale and architectural diversity. In addition, we restrict our evaluation to mathematical reasoning and GUI agent task, which provide structured settings for quantifying performance changes with and without rewards. Although these tasks are well-suited to the questions studied here, they do not cover the full range of domains in which unrewarded exploration may matter. It would therefore be valuable to investigate broader settings, such as open-ended dialogue, embodied control, and code generation, to test whether the gains observed here generalize across a wider spectrum of LLMs' behaviors.

More broadly, our findings suggest that learning in LLM post-training may not depend solely on reward signals. An initial phase of unrewarded exploration can already produce measurable gains and can influence the effectiveness of subsequent training once rewards are introduced. We hope that this work motivates further study of how unrewarded exploration interacts with later optimization, and of when such dynamics may be beneficial across different models, algorithms, and task domains.

## 7 CONCLUSION

In this paper, motivated by Blodgett's rat experiments (Blodgett, 1929), we begin our research by asking *whether an initial phase of training without rewards (i.e., neither external verifier rewards nor internally constructed rewards) produce performance gains, and can subsequent training with rewards introduced after such exploration ultimately yield better performance than training with rewards throughout?* We conduct extensive experiments across task domains, model families, and model sizes, demonstrating that LLMs achieve measurable performance gains when trained without rewards, and continue to achieve higher performance gains after rewards are introduced than training with rewards throughout. Beyond our empirical findings, we also provide theoretical analyses of our experimental setup to explain why LLMs achieve performance gains when trained without rewards.

By connecting a classic construct from cognitive psychology to the training dynamics of contemporary LLMs, our work suggests a potential research direction for future studies to investigate how such a two-stage training paradigm beginning with an unrewarded exploration phase may improve LLMs' performance, especially at much larger scales and across diverse domains.

### AI USE STATEMENT

In this work, we used generative AI tools for "assist with translation". We have not used generative AI tools for "assist in the writing of proofs, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, interpret results", and "generate synthetic datasets, clean and reformat dataset" are not applicable to this work. Additionally, we used generative AI tools for "draft parts of a research paper, summarize or analyse existing literature, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature". We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

Experimental details about training and evaluation are provided in both Section 4.3 and Section 4.4. Our training code is implemented based on verl framework (https://github.com/verl-project/verl). All code is anonymously available at https://anonymous.4open.science/r/11851-1DB9.

## REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016.

Hugh Carlton Blodgett. The effect of the introduction of reward upon the maze performance of rats. *University of California publications in psychology*, 1929.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by Random Network Distillation. In *International Conference on Learning Representations*, 2019.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Guozhi Wang, Dingyu Zhang, Shuai Ren, and Hongsheng Li. Amex: Android multi-annotation expo dataset for mobile gui agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Wentong Chen, Junbo Cui, Jinyi Hu, Yujia Qin, Junjie Fang, Yue Zhao, Chongyi Wang, Jun Liu, Guirong Chen, Yupeng Huo, et al. Guicourse: From general vision language model to versatile gui agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 2025.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems*, 2021.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision*, 2024.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 2011.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *Advances in neural information processing systems*, 2022.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025.

Mengqi Li, Lei Zhao, Anthony Man-Cho So, Ruoyu Sun, and Xiao Li. A model can help itself: Reward-free self-training for llm reasoning. *arXiv preprint arXiv:2510.18814*, 2026.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents. In *Advances in Neural Information Processing Systems*, 2024.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025.

Pierre-Yves Oudeyer, Frederic Kaplan, and Verena Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 2007.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, 2024.

Abhinav Kumar Rastogi, Blaise Aguera y Arcas, Chongyang Bai, Jindong (JD) Chen, Srinivas Kumar Sunkara, Xiaoxue Zang, and Ying Xu. Uibert: Learning generic multimodal representations for ui understanding. In *the 30th International Joint Conference on Artificial Intelligence*, 2021.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2010.

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

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

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. Os-atlas: Foundation action model for generalist gui agents. In *International Conference on Learning Representations*, 2025.

Jian Xiong, Jingbo Zhou, Jingyong Ye, Qiang Huang, and Dejing Dou. AAPO: Enhancing the reasoning capabilities of LLMs with advantage margin. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *Advances in Neural Information Processing Systems*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. Star: Self-taught reasoner bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2024.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Proceedings of the Conference On Language Modeling*, 2025.

Le Zhang, Yixiong Xiao, Xinjiang Lu, Jingjia Cao, Yusai Zhao, Jingbo Zhou, Lang An, Zikan Feng, Wanxiang Sha, Yu Shi, Congxi Xiao, Jian Xiong, Yankai Zhang, Hua Wu, and Haifeng Wang. Omegause: Building a general-purpose gui agent for autonomous task execution. *arXiv preprint arXiv:2601.20380*, 2026a.

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026b.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems*, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, 2026.

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

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

# A PROOF OF THEOREMS

**Definition 2.** *(Latent utility and sequence utility). Let $R(x, y) \in [0, 1]$ denote the terminal task score of a complete output $y$ for input $x$, such as exact-match correctness for mathematical reasoning or task success for GUI control. Fix a continuation policy $\bar{\pi}$ that is independent of the optimization variable $\theta$. For a token state $s = (x, y_{<t})$ and a candidate next token $a$, define the latent utility by*

$$u^\star(s, a) := \mathbb{E}_{y_{>t} \sim \bar{\pi}(\cdot|x, y_{<t}a)} \big[R\big(x, (y_{<t}, a, y_{>t})\big)\big]. \tag{1}$$

*Thus, $u^\star(s, a)$ is the expected eventual task quality of taking token $a$ at state $s$ and then completing the remainder under the fixed continuation policy $\bar{\pi}$. We call it latent because it is not observed by the algorithm during unrewarded exploration and is introduced only for analysis.*

*For a complete output $y = (y_1, \ldots, y_T)$, define the latent sequence utility $U^\star(x, y) := \sum_{t=1}^{T} u^\star(x, y_{<t}, y_t)$.*

*Accordingly, for any policy $\pi$, define the performance functional*

$$J(\pi) := \mathbb{E}_{x \sim D,\ y \sim \pi(\cdot|x)} \big[U^\star(x, y)\big]. \tag{2}$$

**Theorem 1** (Performance improvement without rewards in sparse setting) *Consider an autoregressive language model policy $\pi_\theta$ that defines, for each input prompt $x$ and token prefix $y_{<t}$, a conditional distribution$\pi_\theta(y_t \mid x, y_{<t})$ over the next token $y_t$. The model is trained with Group Relative Policy Optimization where the per-token advantage is fixed to $A_{i,t} \equiv 1$, ratio clipping is applied with parameter $\varepsilon > 0$, and a KL trust-region penalty of strength $\beta > 0$ is included. Let the reference policy be $\pi_{ref}(y_t \mid x, y_{<t})$, and let the proposal distribution be $\pi_{old}(y_t \mid x, y_{<t})$. Assume that the likelihood ratio $h(x, y_{<t}, y_t) = \frac{\pi_{old}(y_t|x,y_{<t})}{\pi_{ref}(y_t|x,y_{<t})}$ is nondecreasing in the latent utility $u^\star(x, y_{<t}, y_t)$ for every input $x$, every prefix $y_{<t}$, and every token $y_t$.*

*Then the GRPO update produces an updated conditional policy of the closed form*

$$\pi^\star_\theta(y_t \mid x, y_{<t}) \;=\; \min\Big((1+\varepsilon)\,\pi_{\text{old}}(y_t \mid x, y_{<t}),\;\; \tau(x, y_{<t})\,\pi_{\text{ref}}(y_t \mid x, y_{<t})\Big), \tag{3}$$

*where $\tau(x, y_{<t}) > 0$ is the unique normalizing constant satisfying $\sum_{y_t \in \mathcal{V}} \pi^\star_\theta(y_t \mid x, y_{<t}) = 1$. Moreover, under the monotone-likelihood-ratio condition, the expected performance*

$$J(\theta) \;=\; \mathbb{E}_{x \sim \mathcal{D}}\, \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[ \sum_{t=1}^{|y|} u^\star(x, y_{<t}, y_t) \right] \tag{4}$$

*satisfies $J(\pi^\star_\theta) \;\geq\; J(\pi_{ref})$. Thus even with no scalar reward signal and with constant advantage, ratio-clipped GRPO with KL regularization yields monotone improvement in the expected utility.*

*Proof.* We work at a fixed token state $s = (x, y_{<t})$ and use Definition 2 throughout. In particular, $u^\star(s, a)$ denotes the expected terminal task score obtained by taking token $a$ at state $s$ and then rolling out the remaining suffix under the fixed continuation policy $\bar{\pi}$. The unrewarded GRPO update itself does not observe $u^\star$; it only optimizes the clipped likelihood-ratio objective with $A_{i,t} \equiv 1$. The role of $u^\star$ is only to define the comparison functional

$$J(\pi) = \mathbb{E}_{x \sim D,\ y \sim \pi(\cdot|x)}[U^\star(x, y)]. \tag{5}$$

The model is a conditional distribution $\pi_\theta(y_t \mid x, y_{<t})$ over tokens in an output sequence $y = (y_1, \ldots, y_T)$ given an input $x$. The autoregressive factorization and log-likelihood decomposition are

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}), \qquad \log \pi_\theta(y \mid x) = \sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t}). \tag{6}$$

For each $x$ we sample a group $G = \{y^{(1)}, \ldots, y^{(K)}\}$ of complete outputs from a fixed proposal policy $\pi_{\text{prop}}(y \mid x)$ (for example a frozen copy of $\pi_{\text{old}}(y \mid x)$). In the setting of interest, the advantage

is set identically to one at every token. Keeping ratio clipping and a direct KL regularizer, the GRPO training objective (ignoring length-normalization constants that do not affect the derivations) is

$$\mathcal{L}_{\text{A=1}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, G\sim\pi_{\text{prop}}^{K}}\left[\sum_{i=1}^{K}\sum_{t=1}^{|y^{(i)}|}\min\Big(\rho_{i,t}(\theta),\ \text{clip}\left(\rho_{i,t}(\theta);\,1-\varepsilon,\,1+\varepsilon\right)\Big)\right] \\ -\ \beta\,\mathbb{E}_{x,t}\Big[\text{KL}\big(\pi_\theta(\cdot\mid x,y_{<t})\,\|\,\pi_{\text{ref}}(\cdot\mid x,y_{<t})\big)\Big], \tag{7}$$

where the per-token importance ratio and its log-increment are

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_t^{(i)}\mid x,y_{<t}^{(i)})}{\pi_{\text{prop}}(y_t^{(i)}\mid x,y_{<t}^{(i)})} = \exp\Big\{\Delta_{i,t}(\theta)\Big\}, \\ \Delta_{i,t}(\theta) = \log\pi_\theta(y_t^{(i)}\mid x,y_{<t}^{(i)}) - \log\pi_{\text{prop}}(y_t^{(i)}\mid x,y_{<t}^{(i)}). \tag{8}$$

The clipping operator is the standard two-sided clamp

$$\text{clip}(\rho;1-\varepsilon,1+\varepsilon) = \min\big\{\max(\rho,1-\varepsilon),\,1+\varepsilon\big\}. \tag{9}$$

With $A\equiv 1$, the inner minimum becomes $\min(\rho_{i,t}(\theta),\,1+\varepsilon)$ at every token. This enables a per-state rewriting. For any fixed token state $(x,y_{<t})$ with discrete vocabulary $\mathcal{V}$ and any constant $c>0$,

$$\sum_{y_t\in\mathcal{V}}\pi_{\text{prop}}(y_t\mid x,y_{<t})\,\min\Big(\frac{\pi_\theta(y_t\mid x,y_{<t})}{\pi_{\text{prop}}(y_t\mid x,y_{<t})},\,c\Big) = \sum_{y_t\in\mathcal{V}}\min\Big(\pi_\theta(y_t\mid x,y_{<t}),\,c\,\pi_{\text{prop}}(y_t\mid x,y_{<t})\Big). \tag{10}$$

Setting $c=1+\varepsilon$ and writing the KL penalty at the same state yields, for each $(x,y_{<t})$, the concave per-state objective

$$\ell_{x,y_{<t}}(\pi) = \sum_{y_t\in\mathcal{V}}\min\big(\pi(y_t\mid x,y_{<t}),\,(1+\varepsilon)\,\pi_{\text{prop}}(y_t\mid x,y_{<t})\big) \\ -\ \beta\sum_{y_t\in\mathcal{V}}\pi(y_t\mid x,y_{<t})\log\frac{\pi(y_t\mid x,y_{<t})}{\pi_{\text{ref}}(y_t\mid x,y_{<t})}, \tag{11}$$

to be maximized over the simplex $\{\pi(\cdot\mid x,y_{<t})\in\Delta^{|\mathcal{V}|-1}\}$. The map $\pi\mapsto\sum_{y_t}\min(\pi,\text{const})$ is concave coordinatewise and $-\beta\,\text{KL}(\pi\|\pi_{\text{ref}})$ is concave in its first argument, so $\ell_{x,y_{<t}}$ is concave.

Introduce a Lagrange multiplier $\lambda_{x,y_{<t}}$ for the constraint $\sum_{y_t}\pi(y_t\mid x,y_{<t})=1$ and nonnegativity multipliers $\mu_{x,y_{<t}}(y_t)\ge 0$. The subgradient of the capped term is

$$\partial_{\pi(y_t|x,y_{<t})}\,\min\big(\pi(y_t\mid x,y_{<t}),(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t})\big) = \begin{cases}\{1\}, & \pi(y_t\mid x,y_{<t})<(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t}),\\ [0,1], & \pi(y_t\mid x,y_{<t})=(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t}),\\ \{0\}, & \pi(y_t\mid x,y_{<t})>(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t}).\end{cases} \tag{12}$$

Hence the KKT stationarity condition states that there exists $\xi_{x,y_{<t}}(y_t)\in[0,1]$ with

$$\xi_{x,y_{<t}}(y_t)\cdot\mathbf{1}\{\pi(y_t\mid x,y_{<t})\ \le(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t})\}\ +\ 0\cdot\mathbf{1}\{\pi(y_t\mid x,y_{<t}) \\ >(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t})\} \\ -\ \beta\Big(\log\tfrac{\pi(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})}+1\Big)\ -\ \lambda_{x,y_{<t}}\ +\ \mu_{x,y_{<t}}(y_t)=0, \tag{13}$$

with complementary slackness $\mu_{x,y_{<t}}(y_t)\,\pi(y_t\mid x,y_{<t})=0$. Whenever $\pi(y_t\mid x,y_{<t})>0$ we have $\mu_{x,y_{<t}}(y_t)=0$. In the interior region where the cap is inactive,

$$\pi(y_t\mid x,y_{<t}) < (1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t}), \tag{14}$$

the subgradient equals equation 6 and equation 13 gives

$$1-\beta\Big(\log\tfrac{\pi(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})}+1\Big)-\lambda_{x,y_{<t}}=0 \quad\Rightarrow\quad \pi(y_t\mid x,y_{<t})=\tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t\mid x,y_{<t}), \tag{15}$$

where $\tau_{x,y_{<t}}\equiv\exp\big(\frac{1-\lambda_{x,y_{<t}}}{\beta}-1\big)$. Combining the interior formula with the active-cap condition $\pi(y_t\mid x,y_{<t})=(1+\varepsilon)\pi_{\text{prop}}(y_t\mid x,y_{<t})$ yields the single closed-form expression for the maximizer at state $(x,y_{<t})$:

$$\pi_\theta^\star(y_t\mid x,y_{<t})\ =\ \min\Big((1+\varepsilon)\,\pi_{\text{prop}}(y_t\mid x,y_{<t}),\ \ \tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t\mid x,y_{<t})\Big),\sum_{y_t\in\mathcal{V}}\pi_\theta^\star(y_t\mid x,y_{<t})=1\,. \tag{16}$$

Strict concavity of $\ell_{x,y_{<t}}$ in the density ratios $\log \frac{\pi}{\pi_{\text{ref}}}$ (via the KL term) ensures the uniqueness of $\tau_{x,y_{<t}} > 0$.

It is convenient to partition the vocabulary at state $(x, y_{<t})$ according to whether the cap is active:

$$S_{x,y_{<t}} = \{y_t : (1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}) \;\le\; \tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t \mid x, y_{<t})\}, \qquad T_{x,y_{<t}} = \mathcal{V} \setminus S_{x,y_{<t}}. \tag{17}$$

By construction,

$$\pi^\star_\theta(y_t \mid x, y_{<t}) = \begin{cases} (1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}), & y_t \in S_{x,y_{<t}}, \\ \tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t \mid x, y_{<t}), & y_t \in T_{x,y_{<t}}, \end{cases} \tag{18}$$

and summing the two pieces gives the exact mass-balance identity

$$\sum_{y_t \in S_{x,y_{<t}}} \Big((1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}) - \pi_{\text{ref}}(y_t \mid x, y_{<t})\Big) + \sum_{y_t \in T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}(y_t \mid x, y_{<t}) \;=\; 0. \tag{19}$$

We now relate the update equation 15 to the latent utility. Let $u^\star(x, y_{<t}, y_t)$ be the (unobserved) tokenwise utility. The expected performance is

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[\sum_{t=1}^{|y|} u^\star(x, y_{<t}, y_t)\Big] = \mathbb{E}_{x,y_{<t}} \sum_{y_t \in \mathcal{V}} \pi_\theta(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t). \tag{20}$$

Assume the monotone-likelihood-ratio condition: for every $(x, y_{<t})$ and any tokens $y_t^{(1)}, y_t^{(2)}$,

$$u^\star(x, y_{<t}, y_t^{(1)}) \;\ge\; u^\star(x, y_{<t}, y_t^{(2)}) \quad\Rightarrow\quad \frac{\pi_{\text{prop}}(y_t^{(1)} \mid x, y_{<t})}{\pi_{\text{ref}}(y_t^{(1)} \mid x, y_{<t})} \;\ge\; \frac{\pi_{\text{prop}}(y_t^{(2)} \mid x, y_{<t})}{\pi_{\text{ref}}(y_t^{(2)} \mid x, y_{<t})}. \tag{21}$$

Define the likelihood ratio

$$h(x, y_{<t}, y_t) = \frac{\pi_{\text{prop}}(y_t \mid x, y_{<t})}{\pi_{\text{ref}}(y_t \mid x, y_{<t})} \tag{22}$$

and the tilt

$$w_{x,y_{<t}}(y_t) = \frac{\pi^\star_\theta(y_t \mid x, y_{<t})}{\pi_{\text{ref}}(y_t \mid x, y_{<t})} = \min\Big(\tau_{x,y_{<t}},\; (1+\varepsilon)\,h(x, y_{<t}, y_t)\Big). \tag{23}$$

Both $w_{x,y_{<t}}(y_t)$ and $u^\star(x, y_{<t}, y_t)$ are nondecreasing functions of the common scalar $h(x, y_{<t}, y_t)$. Chebyshev's association inequality under the base measure $\pi_{\text{old}}(\cdot \mid x, y_{<t})$ therefore gives

$$\begin{aligned} \sum_{y_t \in \mathcal{V}} \pi^\star_\theta(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t) &= \sum_{y_t \in \mathcal{V}} \pi_{\text{old}}(y_t \mid x, y_{<t})\, w_{x,y_{<t}}(y_t)\, u^\star(x, y_{<t}, y_t) \\ &\ge \Big(\sum_{y_t} \pi_{\text{old}}\, w_{x,y_{<t}}\Big)\Big(\sum_{y_t} \pi_{\text{old}}\, u^\star\Big) = \sum_{y_t \in \mathcal{V}} \pi_{\text{old}}(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t), \end{aligned} \tag{24}$$

because $\sum_{y_t} \pi_{\text{old}}(y_t \mid x, y_{<t})\, w_{x,y_{<t}}(y_t) = \sum_{y_t} \pi^\star_\theta(y_t \mid x, y_{<t}) = 1$. Averaging over $(x, y_{<t})$ yields the global monotone improvement

$$J(\pi^\star_\theta) \;\ge\; J(\pi_{\text{old}}). \tag{25}$$

The same conclusion can be expressed in the partition language. Using equation 18-equation 19, the per-state change in performance is

$$\begin{aligned} \Delta J_{x,y_{<t}} &= \sum_{y_t \in \mathcal{V}} \Big(\pi^\star_\theta(y_t \mid x, y_{<t}) - \pi_{\text{old}}(y_t \mid x, y_{<t})\Big)\, u^\star(x, y_{<t}, y_t) \\ &= \sum_{y_t \in S_{x,y_{<t}}} \Big((1+\varepsilon)\pi_{\text{prop}} - \pi_{\text{ref}}\Big)u^\star + \sum_{y_t \in T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}\, u^\star, \end{aligned} \tag{26}$$

where we have abbreviated the arguments to reduce clutter in the last line. Define the nonnegative transfer magnitude

$$\begin{aligned} M_{x,y_{<t}} &\equiv \sum_{y_t \in T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}(y_t \mid x, y_{<t}) \\ &= -\sum_{y_t \in S_{x,y_{<t}}} \Big((1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t}) - \pi_{\text{ref}}(y_t \mid x, y_{<t})\Big) \;\ge\; 0. \end{aligned} \tag{27}$$

Introduce the increment-weighted conditional means

$$\bar{u}^{(+)}_{S_{x,y_{<t}}} = \frac{\sum_{y_t \in S_{x,y_{<t}}} \Big(\pi_{\text{ref}}(y_t \mid x, y_{<t}) - (1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t})\Big)\, u^\star(x, y_{<t}, y_t)}{\sum_{y_t \in S_{x,y_{<t}}} \Big(\pi_{\text{ref}}(y_t \mid x, y_{<t}) - (1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t})\Big)}, \tag{28}$$

$$\bar{u}^{(-)}_{T_{x,y_{<t}}} = \frac{\sum_{y_t \in T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t)}{\sum_{y_t \in T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}(y_t \mid x, y_{<t})}. \tag{29}$$

Then equation 26 can be rewritten as

$$\Delta J_{x,y_{<t}} = -M_{x,y_{<t}}\big(\bar{u}^{(+)}_{S_{x,y_{<t}}} - \bar{u}^{(-)}_{T_{x,y_{<t}}}\big). \tag{30}$$

On $S_{x,y_{<t}}$ the defining inequality $(1+\varepsilon)\pi_{\text{prop}} \le \tau_{x,y_{<t}}\pi_{\text{ref}}$ is equivalent to $h(x, y_{<t}, y_t) \le \tau_{x,y_{<t}}/(1+\varepsilon)$, while the reverse holds on $T_{x,y_{<t}}$. Under the monotone-likelihood-ratio condition equation 24, the conditional mean of $u^\star$ on $T_{x,y_{<t}}$ is at least as large as on $S_{x,y_{<t}}$, which implies $\bar{u}^{(-)}_{T_{x,y_{<t}}} \ge \bar{u}^{(+)}_{S_{x,y_{<t}}}$. Plugging this into equation 30 gives $\Delta J_{x,y_{<t}} \ge 0$. Averaging once more over $(x, y_{<t})$ recovers $J(\pi^\star_\theta) \ge J(\pi_{\text{ref}})$.

Finally, the tokenwise KL penalty admits an unbiased single-sample estimator using $\psi(r) = r - \log r - 1$. For any $(x, y_{<t})$,

$$\mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid x, y_{<t})}\left[\psi\Big(\frac{\pi_{\text{ref}}(y_t \mid x, y_{<t})}{\pi_\theta(y_t \mid x, y_{<t})}\Big)\right] = \mathrm{KL}\big(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \pi_{\text{ref}}(\cdot \mid x, y_{<t})\big), \tag{31}$$

and $\psi(r) \ge 0$ for all $r > 0$ since $\psi''(r) = 1/r > 0$ and $\psi(1) = 0$. This validates the KL term used in equation 7. In small-step stochastic ascent, if $\theta^+ = \theta + \eta\hat{g}$ with $\mathbb{E}[\hat{g}] = \nabla_\theta \mathcal{L}_{\text{A=1}}(\theta)$ and bounded second moment, standard smoothness arguments imply an expected increase in the surrogate. Translating to performance via the association argument above yields the first-order estimate

$$\mathbb{E}\big[J(\theta^+) - J(\theta)\big] \approx \eta\, \mathbb{E}_{x,y_{<t}}\Big[\operatorname{Cov}_{y_t \sim \pi_{\text{ref}}(\cdot \mid x, y_{<t})}\big(\min(\tau_{x,y_{<t}}, (1+\varepsilon)h(x, y_{<t}, y_t)),\, u^\star(x, y_{<t}, y_t)\big)\Big] \ge 0, \tag{32}$$

because both arguments of the covariance are nondecreasing in $h(x, y_{<t}, y_t)$.

In summary, with advantage fixed to one and no scalar rewards, the GRPO objective with ratio clipping and a KL trust region reduces at each token state $(x, y_{<t})$ to maximizing the overlap $\sum_{y_t} \min(\pi(y_t \mid x, y_{<t}), (1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t}))$ minus a KL term to $\pi_{\text{ref}}(\cdot \mid x, y_{<t})$. The unique maximizer is the water-filling solution equation 15. Under the monotone-likelihood-ratio condition equation 24, this update increases the expected latent utility, hence $J(\pi^\star_\theta) \ge J(\pi_{\text{ref}})$, completing the proof. □

**Theorem 2** (Performance improvement without rewards in continuous setting) *Fix an input distribution $\mathcal{D}$, an autoregressive language model policy $\pi_\theta(y_t \mid x, y_{<t})$ that admits a density with respect to a common $\sigma$-finite base measure on the token space, a reference policy $\pi_{ref}(y_t \mid x, y_{<t})$ with the same property, and a proposal (teacher) policy $\pi_{\text{prop}}(y_t \mid x, y_{<t})$. Train with Group Relative Policy Optimization in which the per-token advantage is fixed to $A \equiv 1$, the ratio is clipped with parameter $\varepsilon > 0$ so that the per-token factor becomes $\min(\rho, 1+\varepsilon)$, and a KL trust-region penalty of strength $\beta > 0$ is included. Assume that for every token state $(x, y_{<t})$ the likelihood ratio $h(x, y_{<t}, y_t) = \pi_{\text{prop}}(y_t \mid x, y_{<t})/\pi_{ref}(y_t \mid x, y_{<t})$ is comonotone with the latent token utility $u^\star(x, y_{<t}, y_t)$ under the base distribution $\pi_{ref}(\cdot \mid x, y_{<t})$, meaning that $(u^\star(\cdot) - u^\star(\cdot'))(h(\cdot) - h(\cdot')) \ge 0$ almost everywhere. Then the statewise maximizer of the GRPO surrogate has the water-filling form*

$$\pi^\star_\theta(y_t \mid x, y_{<t}) = \min\big((1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}),\ \tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t \mid x, y_{<t})\big),$$
$$\int \pi^\star_\theta(y_t \mid x, y_{<t})\,\mathrm{d}\nu(y_t) = 1,$$

*for a unique normalizing constant $\tau_{x,y_{<t}} > 0$. Moreover, the expected performance*

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{y\sim\pi_\theta(\cdot \mid x)}\Big[\sum_{t=1}^{|y|} u^\star(x, y_{<t}, y_t)\Big] = \mathbb{E}_{x,y_{<t}} \int \pi_\theta(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t)\,\mathrm{d}\nu(y_t)$$

*satisfies $J(\pi_\theta^\star) \geq J(\pi_{ref})$. Hence, even without scalar rewards and with constant advantage, ratio-clipped GRPO with KL regularization yields monotone improvement in expected latent utility in continuous output spaces.*

*Proof.* The proof proceeds in the continuous measure-theoretic setting by replacing sums with integrals and by making all densities explicit with respect to a common $\sigma$-finite base measure. Fix an input $x$ and a token state $(x, y_{<t})$. Let $(\mathcal{Y}, \mathcal{F}, \nu)$ be a measurable token space with a $\sigma$-finite base measure $\nu$ such that the conditional distributions $\pi_\theta(\cdot \mid x, y_{<t})$, $\pi_{\text{ref}}(\cdot \mid x, y_{<t})$, and $\pi_{\text{prop}}(\cdot \mid x, y_{<t})$ admit densities with respect to $\nu$. The autoregressive factorization and the log-likelihood decomposition remain unchanged,

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}), \qquad \log \pi_\theta(y \mid x) = \sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t}). \tag{33}$$

At each $(x, y_{<t})$, complete sequences are proposed from $\pi_{\text{prop}}(\cdot \mid x, y_{<t})$. Because the per-token advantage is set to $A \equiv 1$, the ratio-clipped token factor becomes $\min(\rho, 1+\varepsilon)$ where

$$\begin{aligned} \rho_{i,t}(\theta) &= \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\text{prop}}(y_t^{(i)} \mid x, y_{<t}^{(i)})} = \exp\{\Delta_{i,t}(\theta)\}, \\ \Delta_{i,t}(\theta) &= \log \pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)}) - \log \pi_{\text{prop}}(y_t^{(i)} \mid x, y_{<t}^{(i)}), \end{aligned} \tag{34}$$

and the two-sided clip is $\text{clip}(\rho; 1-\varepsilon, 1+\varepsilon) = \min\{\max(\rho, 1-\varepsilon),\ 1+\varepsilon\}$. Since $\min(\rho, \text{clip}(\rho)) = \min(\rho, 1+\varepsilon)$ holds identically, expectations over sampled tokens can be rewritten as integrals over densities. For any constant $c > 0$ one has the identity

$$\int_{\mathcal{Y}} \pi_{\text{prop}}(y_t \mid x, y_{<t})\ \min\Big(\frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_{\text{prop}}(y_t \mid x, y_{<t})},\, c\Big)\, \mathrm{d}\nu(y_t) = \int_{\mathcal{Y}} \min\Big(\pi_\theta(y_t \mid x, y_{<t}),\, c\,\pi_{\text{prop}}(y_t \mid x, y_{<t})\Big)\, \mathrm{d}\nu(y_t). \tag{35}$$

Setting $c = 1+\varepsilon$ and writing the KL penalty at the same token state as a functional of the density with respect to $\nu$ gives, for each $(x, y_{<t})$, the continuous surrogate

$$\ell_{x,y_{<t}}(\pi) = \int_{\mathcal{Y}} \min\Big(\pi(y_t \mid x, y_{<t}),\, (1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t})\Big)\, \mathrm{d}\nu(y_t) - \beta \int_{\mathcal{Y}} \pi(y_t \mid x, y_{<t}) \log \frac{\pi(y_t \mid x, y_{<t})}{\pi_{\text{ref}}(y_t \mid x, y_{<t})}\, \mathrm{d}\nu(y_t), \tag{36}$$

to be maximized over the probability simplex of densities at $(x, y_{<t})$,

$$\pi(\cdot \mid x, y_{<t}) \geq 0 \text{ almost everywhere}, \qquad \int_{\mathcal{Y}} \pi(y_t \mid x, y_{<t})\, \mathrm{d}\nu(y_t) = 1. \tag{37}$$

The map $a \mapsto \min(a, \alpha)$ is concave for every $\alpha > 0$, integrals preserve concavity, and $-\beta\,\text{KL}(\pi \| \pi_{\text{ref}})$ is strictly concave in its first argument. Therefore $\ell_{x,y_{<t}}$ is a strictly concave functional on the convex set equation 37.

To derive the closed-form solution, introduce the Lagrangian

$$\mathcal{L}(\pi, \lambda_{x,y_{<t}}, \mu_{x,y_{<t}}) = \ell_{x,y_{<t}}(\pi) - \lambda_{x,y_{<t}} \Big( \int_{\mathcal{Y}} \pi(y_t \mid x, y_{<t})\, \mathrm{d}\nu(y_t) - 1 \Big) - \int_{\mathcal{Y}} \mu_{x,y_{<t}}(y_t)\, \pi(y_t \mid x, y_{<t})\, \mathrm{d}\nu(y_t), \tag{38}$$

where $\lambda_{x,y_{<t}} \in \mathbb{R}$ is the multiplier for normalization and $\mu_{x,y_{<t}}(\cdot) \geq 0$ is the multiplier for nonnegativity. Using the pointwise subgradient of the capped term,

$$\partial_\pi\ \min\big(\pi,\ (1+\varepsilon)\pi_{\text{prop}}\big) = \begin{cases} \{1\}, & \pi < (1+\varepsilon)\pi_{\text{prop}}, \\ [0,1], & \pi = (1+\varepsilon)\pi_{\text{prop}}, \\ \{0\}, & \pi > (1+\varepsilon)\pi_{\text{prop}}, \end{cases} \tag{39}$$

and $\partial_\pi \big[ -\beta\, \pi \log(\pi/\pi_{\text{ref}}) \big] = -\beta(\log(\pi/\pi_{\text{ref}}) + 1)$, the KKT stationarity condition states that there exists a measurable $\xi_{x,y_{<t}}(y_t) \in [0,1]$ such that for almost every $y_t$,

$$\xi_{x,y_{<t}}(y_t) \cdot \mathbf{1}\big\{\pi(y_t \mid x, y_{<t}) \leq (1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t})\big\} + 0 \cdot \mathbf{1}\big\{\pi(y_t \mid x, y_{<t}) > (1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t})\big\} - \beta\Big(\log \tfrac{\pi(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})} + 1\Big) - \lambda_{x,y_{<t}} + \mu_{x,y_{<t}}(y_t) = 0, \tag{40}$$

together with complementary slackness $\mu_{x,y_{<t}}(y_t)\, \pi(y_t \mid x, y_{<t}) = 0$. Whenever $\pi(y_t \mid x, y_{<t}) > 0$ we have $\mu_{x,y_{<t}}(y_t) = 0$. On the interior region where the cap is inactive, namely $\pi < (1+\varepsilon)\pi_{\text{prop}}$, the subgradient equals 1 and equation equation 41 reads

$$1 - \beta\Big( \log \tfrac{\pi(y_t|x,y_{<t})}{\pi_{\text{ref}}(y_t|x,y_{<t})} + 1 \Big) - \lambda_{x,y_{<t}} = 0, \tag{41}$$

which is equivalent to the existence of a scalar

$$\tau_{x,y_{<t}} = \exp\Big(\tfrac{1-\lambda_{x,y_{<t}}}{\beta} - 1\Big) > 0 \tag{42}$$

such that

$$\pi(y_t \mid x, y_{<t}) = \tau_{x,y_{<t}}\, \pi_{\text{ref}}(y_t \mid x, y_{<t}) \qquad \text{almost everywhere on the inactive region.} \tag{43}$$

On the active region where $\pi = (1+\varepsilon)\pi_{\text{prop}}$, KKT allows $\xi \in [0,1]$. Combining with equation 43 gives the statewise water-filling solution

$$\begin{gathered}\pi^\star_\theta(y_t \mid x, y_{<t}) = \min\Big((1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}),\ \ \tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t \mid x, y_{<t})\Big) \\ \text{almost everywhere,} \qquad \int_{\mathcal{Y}} \pi^\star_\theta \,\mathrm{d}\nu = 1\,.\end{gathered} \tag{44}$$

Strict concavity of $\ell_{x,y_{<t}}$ in the density ratios $\log \frac{\pi}{\pi_{\text{ref}}}$ (through the KL term) ensures the uniqueness of $\tau_{x,y_{<t}} > 0$. Concretely, define

$$\Phi_{x,y_{<t}}(\tau) = \int_{\mathcal{Y}} \min\Big((1+\varepsilon)\pi_{\text{prop}}(y_t \mid x, y_{<t}),\ \ \tau\,\pi_{\text{ref}}(y_t \mid x, y_{<t})\Big)\,\mathrm{d}\nu(y_t). \tag{45}$$

The map $\tau \mapsto \Phi_{x,y_{<t}}(\tau)$ is continuous and strictly increasing with $\Phi_{x,y_{<t}}(0) = 0$ and $\lim_{\tau\to\infty}\Phi_{x,y_{<t}}(\tau) = (1+\varepsilon)\int \pi_{\text{prop}}\,\mathrm{d}\nu = 1+\varepsilon > 1$. By the intermediate value theorem there exists a unique $\tau_{x,y_{<t}} > 0$ such that $\Phi_{x,y_{<t}}(\tau_{x,y_{<t}}) = 1$, which enforces normalization in equation 44.

To describe the flow of probability mass, partition the token space at state $(x, y_{<t})$ into the measurable sets

$$S_{x,y_{<t}} = \Big\{y_t :\ (1+\varepsilon)\,\pi_{\text{prop}}(y_t \mid x, y_{<t}) \le \tau_{x,y_{<t}}\,\pi_{\text{ref}}(y_t \mid x, y_{<t})\Big\}, \quad T_{x,y_{<t}} = \mathcal{Y}\setminus S_{x,y_{<t}}. \tag{46}$$

Then almost everywhere $\pi^\star_\theta = (1+\varepsilon)\pi_{\text{prop}}$ on $S_{x,y_{<t}}$ and $\pi^\star_\theta = \tau_{x,y_{<t}}\pi_{\text{ref}}$ on $T_{x,y_{<t}}$, and summing these two pieces yields the continuous mass-balance identity

$$\int_{S_{x,y_{<t}}} \Big((1+\varepsilon)\,\pi_{\text{prop}} - \pi_{\text{ref}}\Big)\,\mathrm{d}\nu + \int_{T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}\,\mathrm{d}\nu = 0. \tag{47}$$

Introduce the unobserved token-level utility $u^\star(x, y_{<t}, y_t)$, assumed integrable with respect to $\pi_{\text{ref}}(\cdot \mid x, y_{<t})$. The expected performance is

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[\textstyle\sum_{t=1}^{|y|} u^\star(x, y_{<t}, y_t)\Big] = \mathbb{E}_{x,y_{<t}}\textstyle\int_{\mathcal{Y}} \pi_\theta(y_t \mid x, y_{<t})\,u^\star(x, y_{<t}, y_t)\,\mathrm{d}\nu(y_t). \tag{48}$$

To connect the update in equation 44 to $J$, impose the continuous monotone-likelihood-ratio structure: for each $(x, y_{<t})$ define

$$h(x, y_{<t}, y_t) = \frac{\pi_{\text{prop}}(y_t \mid x, y_{<t})}{\pi_{\text{ref}}(y_t \mid x, y_{<t})}, \tag{49}$$

and assume that $u^\star(x, y_{<t}, \cdot)$ is comonotone with $h(x, y_{<t}, \cdot)$ under $\pi_{\text{ref}}(\cdot \mid x, y_{<t})$, that is,

$$\big(u^\star(x, y_{<t}, y_t) - u^\star(x, y_{<t}, y'_t)\big)\cdot\big(h(x, y_{<t}, y_t) - h(x, y_{<t}, y'_t)\big) \ge 0 \quad \text{for almost every } (y_t, y'_t). \tag{50}$$

Define the tilt

$$w_{x,y_{<t}}(y_t) = \frac{\pi^\star_\theta(y_t \mid x, y_{<t})}{\pi_{\text{ref}}(y_t \mid x, y_{<t})} = \min\Big(\tau_{x,y_{<t}},\ (1+\varepsilon)\,h(x, y_{<t}, y_t)\Big). \tag{51}$$

The map $z \mapsto \min(\tau_{x,y_{<t}}, (1+\varepsilon)z)$ is nondecreasing, and by assumption $u^\star$ is nondecreasing in $h$. Therefore $w_{x,y_{<t}}(\cdot)$ and $u^\star(x, y_{<t}, \cdot)$ are comonotone random variables under the base probability measure $\mathrm{d}\mu_{x,y_{<t}}(y_t) = \pi_{\text{ref}}(y_t \mid x, y_{<t})\,\mathrm{d}\nu(y_t)$. Chebyshev's association inequality in integral form gives

$$\int w_{x,y_{<t}}(y_t)\,u^\star(x, y_{<t}, y_t)\,\mathrm{d}\mu_{x,y_{<t}} \ \ge\ \Big(\int w_{x,y_{<t}}\,\mathrm{d}\mu_{x,y_{<t}}\Big)\Big(\int u^\star\,\mathrm{d}\mu_{x,y_{<t}}\Big). \tag{52}$$

Since $\int w_{x,y_{<t}}\,\mathrm{d}\mu_{x,y_{<t}} = \int \pi_\theta^\star\,\mathrm{d}\nu = 1$, we deduce

$$\int \pi_\theta^\star(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t)\,\mathrm{d}\nu \;\geq\; \int \pi_{\text{ref}}(y_t \mid x, y_{<t})\, u^\star(x, y_{<t}, y_t)\,\mathrm{d}\nu. \tag{53}$$

Averaging over $(x, y_{<t})$ yields the global monotone improvement

$$J(\pi_\theta^\star) \;\geq\; J(\pi_{\text{ref}}). \tag{54}$$

The same conclusion can be expressed in the language of mass transfer. The per-state change in performance is

$$\Delta J_{x,y_{<t}} = \int_{\mathcal{Y}} \Big(\pi_\theta^\star(y_t \mid x, y_{<t}) - \pi_{\text{ref}}(y_t \mid x, y_{<t})\Big)\, u^\star(x, y_{<t}, y_t)\,\mathrm{d}\nu. \tag{55}$$

Using equation 44–equation 47 this decomposes as

$$\Delta J_{x,y_{<t}} = \int_{S_{x,y_{<t}}} \Big((1+\varepsilon)\pi_{\text{prop}} - \pi_{\text{ref}}\Big) u^\star\,\mathrm{d}\nu + \int_{T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}\, u^\star\,\mathrm{d}\nu. \tag{56}$$

Define the nonnegative transfer magnitude

$$M_{x,y_{<t}} = \int_{T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}\,\mathrm{d}\nu = -\int_{S_{x,y_{<t}}} \Big((1+\varepsilon)\pi_{\text{prop}} - \pi_{\text{ref}}\Big)\,\mathrm{d}\nu \;\geq 0, \tag{57}$$

and the increment-weighted conditional means

$$\bar{u}^{(+)}_{S_{x,y_{<t}}} = \frac{\int_{S_{x,y_{<t}}} \Big(\pi_{\text{ref}} - (1+\varepsilon)\pi_{\text{prop}}\Big)\, u^\star\,\mathrm{d}\nu}{\int_{S_{x,y_{<t}}} \Big(\pi_{\text{ref}} - (1+\varepsilon)\pi_{\text{prop}}\Big)\,\mathrm{d}\nu}, \qquad \bar{u}^{(-)}_{T_{x,y_{<t}}} = \frac{\int_{T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}\, u^\star\,\mathrm{d}\nu}{\int_{T_{x,y_{<t}}} \Big(\tau_{x,y_{<t}} - 1\Big)\pi_{\text{ref}}\,\mathrm{d}\nu}. \tag{58}$$

Then

$$\Delta J_{x,y_{<t}} = -\,M_{x,y_{<t}}\big(\bar{u}^{(+)}_{S_{x,y_{<t}}} - \bar{u}^{(-)}_{T_{x,y_{<t}}}\big). \tag{59}$$

By definition, on $S_{x,y_{<t}}$ one has $(1+\varepsilon)\pi_{\text{prop}} \leq \tau_{x,y_{<t}}\pi_{\text{ref}}$, which is equivalent to $h(x, y_{<t}, y_t) \leq \tau_{x,y_{<t}}/(1+\varepsilon)$; the reverse holds on $T_{x,y_{<t}}$. Under the comonotonicity condition equation 50 the conditional mean of $u^\star$ on $T_{x,y_{<t}}$ is at least that on $S_{x,y_{<t}}$, hence $\bar{u}^{(-)}_{T_{x,y_{<t}}} \geq \bar{u}^{(+)}_{S_{x,y_{<t}}}$, and therefore $\Delta J_{x,y_{<t}} \geq 0$. Averaging over $(x, y_{<t})$ recovers equation 54.

It is useful to verify that the tokenwise KL penalty admits an unbiased single-sample estimator in the continuous case. Define $\psi(r) = r - \log r - 1$. For any $(x, y_{<t})$,

$$\int_{\mathcal{Y}} \psi\Big(\frac{\pi_{\text{ref}}(y_t \mid x, y_{<t})}{\pi_\theta(y_t \mid x, y_{<t})}\Big)\,\pi_\theta(y_t \mid x, y_{<t})\,\mathrm{d}\nu(y_t) = \int \Big(\tfrac{\pi_{\text{ref}}}{\pi_\theta} - \log\tfrac{\pi_{\text{ref}}}{\pi_\theta} - 1\Big)\,\pi_\theta\,\mathrm{d}\nu = \mathrm{KL}\big(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \pi_{\text{ref}}(\cdot \mid x, y_{<t})\big), \tag{60}$$

and $\psi(r) \geq 0$ for all $r > 0$ because $\psi''(r) = 1/r > 0$ and $\psi(1) = 0$. This validates the KL term in the surrogate equation 36. In small-step stochastic ascent, if $\theta^+ = \theta + \eta\hat{g}$ with $\mathbb{E}[\hat{g}] = \nabla_\theta$ of the empirical surrogate and bounded second moment, standard smoothness arguments imply an expected increase in the surrogate. Translating to performance via the association argument above yields the first-order estimate

$$\mathbb{E}\big[J(\theta^+) - J(\theta)\big] \;\approx\; \eta\,\mathbb{E}_{x,y_{<t}}\Big[\mathrm{Cov}_{Y_t\sim\pi_{\text{ref}}(\cdot|x,y_{<t})}\big(\min(\tau_{x,y_{<t}}, (1+\varepsilon)h(x, y_{<t}, Y_t)),\; u^\star(x, y_{<t}, Y_t)\big)\Big] \geq 0, \tag{61}$$

because both arguments of the covariance are nondecreasing functions of the common scalar $h(x, y_{<t}, Y_t)$.

In conclusion, in continuous output spaces, provided the reference and proposal policies admit densities with respect to a common base measure and satisfy the comonotone likelihood-ratio structure with the latent utility, GRPO with no reward and constant advantage, equipped with ratio clipping and a KL trust region, reduces at each token state $(x, y_{<t})$ to maximizing the overlap integral $\int \min(\pi, (1+\varepsilon)\pi_{\text{prop}})$ minus a KL divergence, a strictly concave problem with the unique water-filling maximizer $\pi_\theta^\star = \min((1+\varepsilon)\pi_{\text{prop}},\, \tau\,\pi_{\text{ref}})$. This update transfers probability mass from low ratio regions to high ratio regions and, by the integral association inequality together with the mass-balance identity, guarantees a monotone increase of the expected latent utility $J$ without requiring any scalar reward signal. □

# B Extended Experiments

## B.1 Effectiveness beyond GRPO

We further investigate whether a model that has undergone the unrewarded exploration phase can provide a better starting point for other RL methods. Whether the phenomenon observed in GRPO, where using the resulting model after unrewarded exploration as the starting point for RL yields better training results than using RL throughout the entire process, also holds true for other RL methods. We select DAPO (Yu et al., 2025) as another RL baseline for these additional experiments. Detailed experimental results are shown in Table 7. Experimental results show that the phenomenon whereby conducting unrewarded exploration prior to RL training yields additional benefits is not limited to GRPO; the same phenomenon is also observed in DAPO.

Table 7: Ablation study of using different RL method.

| Model | GSM8K | MATH500 | Minerva | OlympiadBench | AIME24 | AMC23 | Avg |
|---|---|---|---|---|---|---|---|
| *Qwen 1.5B Models* | | | | | | | |
| Qwen2.5-1.5B-Instruct | 68.7 | 47.6 | 16.5 | 18.1 | 0.0 | 20 | 28.5 |
| + Rewarded (DAPO) | 74.8 | 54.8 | 19.5 | 19.6 | 3.3 | 30.0 | 33.7 |
| + Unrewarded → Rewarded (DAPO) | 73.4 (-1.4) | 56.0 (+1.2) | 21.7 (+2.2) | 19.7 (+0.1) | 3.3 | 32.5 (+2.5) | **34.4** (+0.7) |
| *Qwen 3B Models* | | | | | | | |
| Qwen2.5-3B-Instruct | 84.9 | 63.0 | 24.3 | 24.4 | 6.7 | 37.5 | 40.1 |
| + Rewarded (DAPO) | 85.2 | 66.0 | 30.1 | 27.6 | 6.7 | 47.5 | 43.9 |
| + Unrewarded → Rewarded (DAPO) | 84.5 (-0.7) | 65.8 (-0.2) | 34.6 (+4.5) | 27.6 | 13.3 (+6.6) | 45.0 (-2.5) | **45.1** (+1.2) |

## B.2 Comparing with Internal Rewards

We further compare whether this type of unrewarded exploration without reward signals performs better than methods that construct rewards based on the models' own information. We selected INTUITOR (Zhao et al., 2026) as the baseline, which trains models using the average KL divergence between a uniform distribution $\mathcal{U}$ over the vocabulary $\mathcal{V}$ and the models' next-token distribution as rewards, while our unrewarded exploration does not require any rewards, neither from an external verifier nor constructed based on the models' internal information. The experimental results in Table 8 show that *Unrewarded → Rewarded* consistently outperforms INTUITOR. Moreover, *Unrewarded* demonstrates superior performance compared to INTUITOR on the 3B model.

Table 8: Comparing with an internally constructed reward baseline.

| Model | GSM8K | MATH500 | Minerva | OlympiadBench | AIME24 | AMC23 | Avg |
|---|---|---|---|---|---|---|---|
| *Qwen 1.5B Models* | | | | | | | |
| Qwen2.5-1.5B-Instruct | 68.7 | 47.6 | 16.5 | 18.1 | 0.0 | 20 | 28.5 |
| + Unrewarded | 68.2 | 48.0 | 17.3 | 19.3 | 0.0 | 32.5 | 30.9 |
| + Unrewarded → Rewarded | 72.2 | 54.2 | 19.5 | 19.4 | 10.0 | 37.5 | **35.5** |
| + INTUITOR | 72.1 | 52.6 | 15.1 | 16.3 | 3.3 | 32.5 | 32.0 |
| *Qwen 3B Models* | | | | | | | |
| Qwen2.5-3B-Instruct | 84.9 | 63.0 | 24.3 | 24.4 | 6.7 | 37.5 | 40.1 |
| + Unrewarded | 85.7 | 61.4 | 25.7 | 27.0 | 6.7 | 50.0 | 42.8 |
| + Unrewarded → Rewarded | 85.6 | 67.6 | 25.7 | 29.3 | 10.0 | 55.0 | **45.5** |
| + INTUITOR | 84.7 | 63.0 | 25.7 | 29.5 | 13.3 | 40.0 | 42.7 |

# C Training Dynamics Comparison

In this section, we provide a detailed comparison of the performance of unrewarded exploration combined with rewarded training, as well as rewarded training alone, as the training budget increases.

Figure 2 and Figure 3 show the changes in the average performance of different model sizes from the Qwen and LLaMA model families across multiple benchmarks for mathematical reasoning task as the training budget increases. For most training scenarios, the performance gains achieved during the unrewarded exploration phase exceed those achieved during the rewarded phase (i.e., the shaded areas in each subfigure). Furthermore, the "unrewarded exploration + rewarded" training consistently achieves better results than rewarded training alone within a training budget equal to or less than that of rewarded training alone. This indicates that the introduction of the unrewarded exploration phase provides a better starting point for subsequent rewarded training.

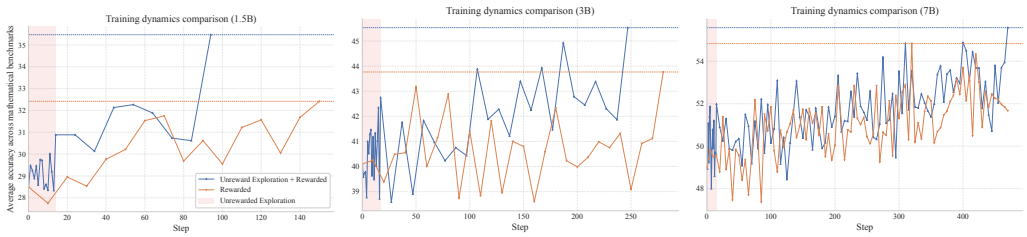


Figure 2: Comparison of training performance on the Qwen series.

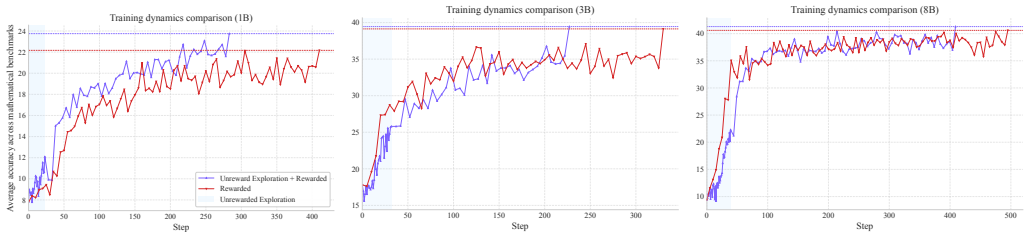


Figure 3: Comparison of training performance on the LLaMA series.

# D More Experimental Results

## D.1 Pass@k Performance

In addition to the pass@1 results from the main experiments, we also include the pass@k experimental results in Table 9, with k ranging from 2 to 32. In most cases, the results obtained by first conducting unrewarded exploration followed by rewarded training are no worse than those obtained through rewarded training throughout. Moreover, unrewarded exploration even shows superior pass@k performance for larger values of k on benchmarks such as MATH500, OlympiadBench, and AMC23 on Qwen2.5-3B-Instruct model. We plot the pass@k curves for the Qwen and LLaMA series on AIME24 and AMC23 benchmarks in Figure 4 and Figure 5, with k ranging from 1 to 1024. For the Qwen series, the results for the 1.5B model on AMC23 show that the "unrewarded exploration + rewarded" training consistently outperforms both the base model and the "unrewarded exploration" training for all values of k, and the "unrewarded exploration" training outperforms the base model at larger values of k; the 3B model on AIME24 shows that the "unrewarded exploration + rewarded" training consistently outperforms the base model for all values of k, and the "unrewarded exploration" training outperforms both the base model and the "unrewarded exploration + rewarded" training for larger values of k. For the LLaMA series, the 1B and 3B models on AIME24 show that "unrewarded exploration" training consistently outperforms the base model; the 1B model on AMC23 shows that both "unrewarded exploration" and "unrewarded exploration + rewarded" training consistently outperform the base model, except for k=1024 for "unrewarded exploration". Some cases also exhibit the same phenomenon observed in (Yue et al., 2025): when k is large, the pass@k performance of models trained by RLVR is inferior to that of the base model, which is also observed in "unrewarded exploration", such as Llama-3.2-3B-Instruct model on AMC23 in Figure 5.

Table 9: Pass@k performance of different training results.

| | K | GSM8K | MATH500 | Minerva | OlympiadBench | AIME24 | AMC23 |
|---|---|---|---|---|---|---|---|
| Qwen2.5-1.5B-Instruct | | | | | | | |
| Rewarded | 2 | 82.0 | 63.1 | 25.0 | 24.7 | 4.7 | 38.2 |
| | 4 | 88.5 | 71.2 | 32.3 | 32.0 | 7.9 | 49.7 |
| | 8 | 92.7 | 77.8 | 39.1 | 39.0 | 12.2 | 61.3 |
| | 16 | 95.2 | 83.4 | 45.9 | 45.6 | 17.8 | 72.6 |
| | 32 | 96.8 | 88.2 | 52.9 | 52.1 | 23.3 | 82.5 |
| Unrewarded | 2 | 79.9 | 60.0 | 23.7 | 22.4 | 4.4 | 34.4 |
| | 4 | 87.5 | 69.2 | 31.7 | 30.0 | 7.3 | 47.0 |
| | 8 | 92.1 | 76.8 | 39.5 | 37.7 | 11.4 | 60.4 |
| | 16 | 94.8 | 83.1 | 46.4 | 45.5 | 17.2 | 72.5 |
| | 32 | 96.4 | 87.8 | 52.6 | 53.6 | 23.3 | 80.0 |
| Unrewarded → Rewarded | 2 | 82.0 | 63.0 | 26.2 | 25.5 | 4.6 | 37.3 |
| | 4 | 88.5 | 71.0 | 33.9 | 32.9 | 7.8 | 48.8 |
| | 8 | 92.8 | 77.6 | 40.8 | 40.0 | 12.6 | 59.6 |
| | 16 | 95.6 | 83.1 | 47.0 | 47.1 | 19.1 | 69.1 |
| | 32 | 97.5 | 88.2 | 52.9 | 54.4 | 26.7 | 77.5 |
| Qwen2.5-3B-Instruct | | | | | | | |
| Rewarded | 2 | 90.8 | 74.3 | 38.0 | 35.5 | 9.2 | 48.6 |
| | 4 | 94.0 | 80.8 | 45.3 | 43.2 | 12.9 | 59.6 |
| | 8 | 95.8 | 85.6 | 51.3 | 50.4 | 17.4 | 68.8 |
| | 16 | 97.0 | 89.0 | 56.5 | 56.6 | 22.9 | 76.3 |
| | 32 | 98.0 | 91.4 | 61.8 | 61.6 | 30.0 | 82.5 |
| Unrewarded | 2 | 90.6 | 73.2 | 31.4 | 34.2 | 7.8 | 49.4 |
| | 4 | 93.9 | 80.3 | 38.5 | 42.2 | 11.0 | 62.0 |
| | 8 | 95.9 | 85.6 | 44.7 | 49.8 | 15.2 | 72.8 |
| | 16 | 97.1 | 89.2 | 50.3 | 56.7 | 21.0 | 80.9 |
| | 32 | 97.8 | 91.8 | 55.9 | 62.8 | 26.7 | 87.5 |
| Unrewarded → Rewarded | 2 | 90.8 | 75.0 | 38.4 | 36.3 | 11.0 | 50.4 |
| | 4 | 93.7 | 81.5 | 45.1 | 43.6 | 15.5 | 60.5 |
| | 8 | 95.6 | 86.2 | 50.8 | 50.2 | 20.3 | 69.3 |
| | 16 | 96.9 | 89.4 | 55.7 | 56.0 | 25.8 | 76.6 |
| | 32 | 97.8 | 91.4 | 61.0 | 61.3 | 33.3 | 82.5 |

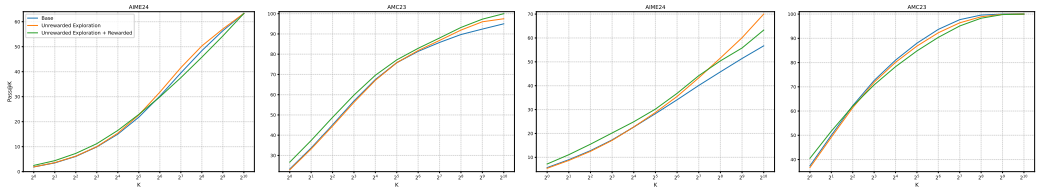


Figure 4: Pass@k performance on AIME24 and AMC23 benchmarks. The two subfigures on the left are Qwen2.5-1.5B-Instruct results. The two subfigures on the right are Qwen2.5-3B-Instruct results.

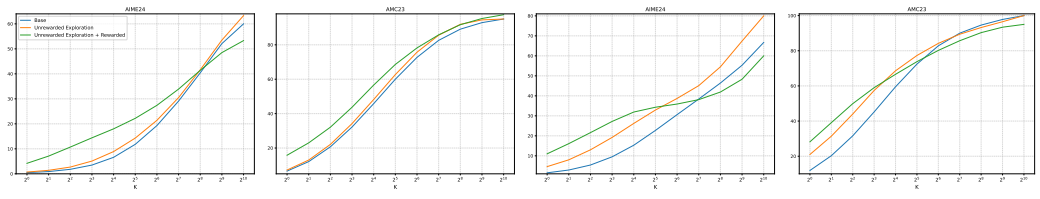


Figure 5: Pass@k performance on AIME24 and AMC23 benchmarks. The two subfigures on the left are Llama-3.2-1B-Instruct results. The two subfigures on the right are Llama-3.2-3B-Instruct results.

### D.2 Training Performance during Unrewarded Exploration

Figure 6 and Figure 7 show more detailed experimental results of unrewarded exploration for our mathematical reasoning task on the Qwen and LLaMA series. As can be seen, during the unrewarded exploration phase, the LLMs' performance improves across all mathematical reasoning benchmarks. However, for smaller model sizes (i.e., 1B, 1.5B, and 3B), a sharp decline in response tokens and performance occurs, which is especially manifest on the LLaMA series models. A similar trend is observed at larger model sizes (7B and 8B), though it occurs later. This is a failure phenomenon observed in mathematical reasoning task, as the training steps further, both the response tokens and performance decline sharply when the model is evaluated on benchmarks. The reasons behind this phenomenon warrant further analysis in future research.

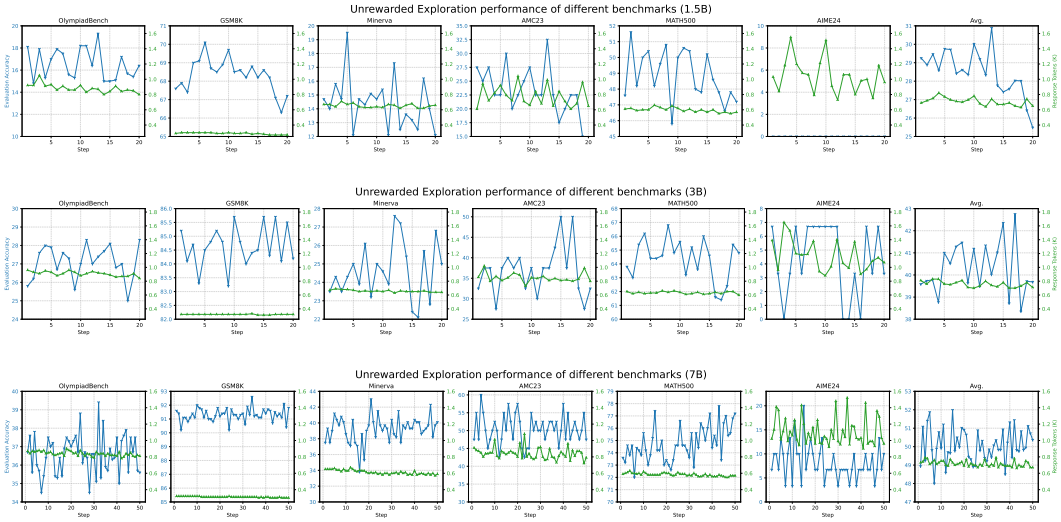


Figure 6: Training performance during unrewarded exploration on the Qwen series.

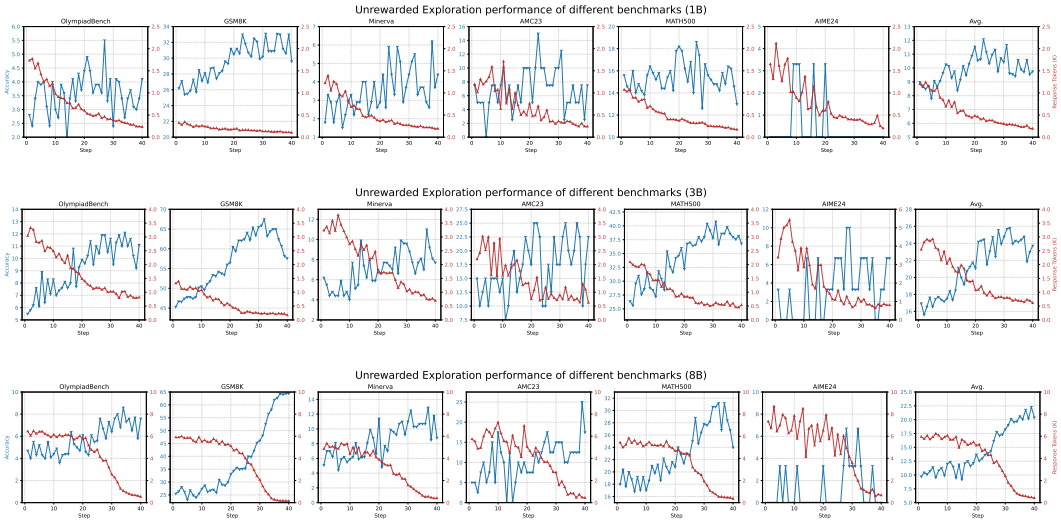


Figure 7: Training performance during unrewarded exploration on the LLaMA series.

# E More Related Work

## E.1 More Training Methods for LLMs

Self-training and self-distillation in LLMs form another closely related line of research. In the LLM literature, self-training typically improves LLMs by converting selected model-generated outputs, such as rationales or solutions, into discrete supervision labels for further fine-tuning. Representative examples include STaR (Zelikman et al., 2024), which iteratively retains self-generated rationales that lead to correct answers, and ReST$^{\text{EM}}$ (Singh et al., 2024), which repeatedly generates, filters, and fine-tunes on self-generated solutions. From the perspective of the learning objective, such self-training methods typically optimize cross-entropy loss over selected discrete pseudo-labels, which can be viewed as learning from hard targets. SePT (Li et al., 2026) shows that LLMs can improve their reasoning performance by sampling low-temperature responses from LLMs and performing SFT on these responses with training temperature set to 1.0 on the mathematical reasoning task. By contrast, self-distillation in LLMs focuses on transferring information through teacher–student distribution matching rather than treating selected generations as direct supervised labels. SSD (Zhang et al., 2026b) shows that LLMs can improve code generation using only their own raw responses by simply performing SFT on these responses. On-policy self-distillation (Hübotter et al., 2026) explicitly trains the student to match the teacher's token-level distribution on the student's own rollouts. Our unrewarded exploration setting differs from both directions. Unlike self-training, unrewarded exploration does not optimize cross-entropy loss on selected pseudo-labels generated by LLMs. It does not convert sampled responses into explicit pseudo-labels for subsequent supervised learning, as in self-training, nor does it train the LLMs to match a teacher policy's output distribution, as in self-distillation. Instead, the LLMs are updated on all of their own samples directly using the objective in Section 3.2 during unrewarded exploration, rather than by fitting hard pseudo-labels or matching a teacher distribution.

Table 10: Comparison between different training methods in LLMs.

| Training Method | Dense Signal | Teacher Model | Verifier | Filtering | (Pseudo) Label |
|---|---|---|---|---|---|
| Self-training | ✗ | ✗ | ✗ | ✓ | ✓ |
| (On-policy) Self-distillation | ✓ | ✓ | ✗ | ✗ | ✓ |
| Reinforcement Learning | ✗ | ✗ | ✓ | ✗ | ✓ |
| Unrewarded Exploration | ✗ | ✗ | ✗ | ✗ | ✗ |

Table 10 compares different training methods in LLMs. Here, *Dense Signal* denotes distributional supervision, *Teacher Model* indicates whether the training process uses a teacher model, *Verifier* indicates whether answer correctness or reward must be verified, *Filtering* indicates whether the generated samples are selected before training, while *(Pseudo) Label* indicates whether the initial stage relies on labeled data or whether pseudo-labels are used. Under these definitions, self-training typically uses verifier-based filtering and converts selected generations into discrete supervision labels, whereas self-distillation relies on teacher-distribution matching. On-policy self-distillation further introduces privileged information into the teacher context. Reinforcement learning and unrewarded exploration are shown separately because unrewarded exploration in our paper is a special case of the RL objective with constant advantage. In unrewarded exploration, we do not need a verifier since there is no reward assignment.

## E.2 Intrinsic-Motivation and Novelty-Driven Exploration

Exploration without extrinsic task rewards has a long history in reinforcement learning, including artificial curiosity and intrinsic-motivation methods (Schmidhuber, 2010; Oudeyer et al., 2007; Pathak et al., 2017), count-based exploration (Bellemare et al., 2016), and random network distillation (Burda et al., 2019). These methods remove or reduce reliance on external task rewards but still introduce behavior-dependent exploration signals: for example, Schmidhuber (2010) uses curiosity reinforcement derived from prediction-model improvement, Oudeyer et al. (2007) use learning progress as an internal reward, Pathak et al. (2017) use prediction error as intrinsic reward, Bellemare et al. (2016) convert pseudo-counts into exploration bonuses, and Burda et al. (2019) use random-network prediction error as an exploration bonus. Similarly, novelty search abandons the task objective but

explicitly searches for behavioral novelty, thereby still differentiating behaviors according to a novelty criterion (Lehman & Stanley, 2011). In contrast, unrewarded exploration uses neither external task rewards nor behavior-dependent intrinsic rewards or novelty-based exploration signals. All sampled responses receive the same constant advantage and are therefore not differentiated by correctness, prediction error, novelty, surprise, or informativeness.

## F DIFFERENCE BETWEEN TWO MATHEMATICAL DATASETS

Here, we provide a detailed explanation of the differences between simplelr_qwen_level3to5 and simplelr_abel_level3to5. The only difference between the two datasets lies in the output-format requirements specified in the prompts. The simplelr_qwen_level3to5 dataset requires the model to output the final answer within the \box{} tag, while the simplelr_abel_level3to5 dataset has no output format requirements. Below are two examples from simplelr_qwen_level3to5 and simplelr_abel_level3to5.

Example from simplelr_qwen_level3to5 (training dataset for the Qwen series)

Mrs. Gable's third grade class is on a field trip to the beach. For lunch, Mrs. Gable brought 20 lunches for the 20 students in her class. She included a pack of animal crackers in each lunch bag for dessert. Each pack of animal crackers contained 10 animal crackers. If 2 students did not eat their animal crackers, how many animal crackers were eaten in total among Mrs. Gable's students?
Please reason step by step, and put your final answer within \\boxed{{}}.

Example from simplelr_abel_level3to5 (training dataset for the LLaMA series)

Mrs. Gable's third grade class is on a field trip to the beach. For lunch, Mrs. Gable brought 20 lunches for the 20 students in her class. She included a pack of animal crackers in each lunch bag for dessert. Each pack of animal crackers contained 10 animal crackers. If 2 students did not eat their animal crackers, how many animal crackers were eaten in total among Mrs. Gable's students?
Answer:
Let's think step by step.