Reasoning Warm-up: Scaling Label-free RL via Verifiable Surrogate Rewards

Jun Nie[1,2] **Bo Han**[2†] **Jiaqi Fan**[3] **Yonggang Zhang**[4†]
Xinmei Tian[1] **Dahai Yu**[3] **Michael K. Ng**[5]

[1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition,
University of Science and Technology of China
[2]TMLR Group, Hong Kong Baptist University [3]TCL Corporate Research (HK) Co., Ltd.
[4]The Hong Kong University of Science and Technology, HKGAI
[5]Department of Mathematics, Hong Kong Baptist University

Abstract

Improving the reasoning ability of large language models (LLMs) without ground-truth supervision remains a central challenge in reinforcement learning (RL). Recent label-free RL methods typically rely on internal pseudo-rewards such as self-consistency or majority voting. However, for complex multi-step reasoning, these signals can be systematically misleading: models may repeatedly produce mutually consistent but incorrect trajectories, causing optimization to favor frequency over correctness. In this work, we propose reasoning warm-up reinforcement learning, a label-free RL framework motivated by an empirical phenomenon we call **Reasoning Warm-up**. We observe that when a model successfully completes a deterministic auxiliary task with an exact verifier, its success probability on a subsequent complex reasoning task increases substantially. This coupling suggests that auxiliary-task verification can serve as a useful surrogate signal for selecting and optimizing reasoning trajectories even when the main task itself is not directly verifiable. Based on this observation, we incorporate verifiable auxiliary tasks into both generation and optimization. During generation, the auxiliary task provides a verifiable warm-up prefix; during training, its verification outcome is used to score candidate trajectories for policy optimization. Experiments on multiple benchmarks demonstrate the effectiveness of our method.

1 Introduction

Recent advances in large language models (LLMs) have shown that reasoning ability can be substantially improved not only by scaling pre-training, but also by post-training optimization. Chain-of-thought prompting enables models to expose intermediate reasoning steps and has become a standard mechanism for eliciting mathematical and symbolic reasoning (Wei et al., 2022). Reinforcement learning from human or verifiable feedback further demonstrates that carefully designed reward signals can significantly shape model behavior after pre-training (Christiano et al., 2017; Ouyang et al., 2022; Schulman et al., 2017). In mathematical reasoning, this paradigm is particularly appealing because final answers can often be checked exactly, making the domain a natural testbed for reinforcement learning from verifiable rewards (Lightman et al., 2024; Shao et al., 2024).

The success of verifiable-reward training naturally raises a more stringent question: can reasoning be improved when ground-truth answers are not used as reward labels during optimization? This

[†]Correspondence to Bo Han (bhanml@comp.hkbu.edu.hk) and Yonggang Zhang (zhangyg@ust.hk).
Code: https://github.com/junnie00/RWR

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

The model is required to complete two tasks. *Task 1* is a deterministic auxiliary task: read the sequence $820, 937, 820, 515, 123, 755, 783, 353, 544$, count from the end, and output the 6th number. The required output format is `<aux>...</aux>`. *Task 2* is the main mathematical problem: for rectangular boxes with surface area 54 and volume 23, find $p + q$, where $r^2 = p/q$. The required output format is `\boxed{...}`.

Abridged model output.

Task 1: Auxiliary task. The model counts from the end: 544, 353, 783, 755, 123, 515.
It then outputs:

`<aux>515</aux>` ✓

Task 2: Main task. The model sets the box dimensions as a, b, c, uses $ab + bc + ca = 27$ and $abc = 23$, and reduces the extremal case to $a = b$. It solves $a^3 - 27a + 46 = 0$, obtains $a = b = 2$ and $c = 23/4$, and computes $a^2 + b^2 + c^2 = 657/16$. Hence $r^2 = 657/64$, so the model outputs:

`\boxed{721}` ✓

The auxiliary answer is verified, and the subsequent main-task answer is correct.

Abridged model output.

Task 1: Auxiliary task. The model miscounts the same sequence and claims that the 6th number from the end is 755. It outputs:

`<aux>755</aux>` ×

Task 2: Main task. The model starts with $abc = 23$ and $ab + bc + ca = 27$, but then follows an invalid path. It assumes $a = b = 1$ and $c = 23$, leading to $a + b + c = 25$. It then computes $a^2 + b^2 + c^2 = 25^2 - 54 = 571$, obtains $r^2 = 571/4$, and outputs:

`\boxed{575}` ×

The auxiliary answer is incorrect, and the subsequent main-task answer is also incorrect.

Figure 1: Under the same auxiliary-augmented prompt, the trajectory with a correct auxiliary answer also reaches the correct main-task answer, whereas the trajectory with an incorrect auxiliary answer subsequently fails on the main task. Model outputs are abridged for readability.

question is connected to a broader line of self-improvement and AI-feedback training, where models are improved using unlabeled data, model-generated rationales, AI feedback, or self-generated rewards rather than direct human or task labels (Huang et al., 2023; Bai et al., 2022; Lee et al., 2023; Yuan et al., 2024). Such a setting is meaningful for both practical and scientific reasons. Practically, many reasoning tasks do not come with cheap and reliable verifiers, and collecting task-specific labels at scale can be costly or infeasible. Scientifically, withholding direct task labels provides a controlled stress test for label-free optimization: it allows us to examine whether useful learning signals can be constructed from sources other than answer correctness. The central challenge therefore shifts from answer verification to reward construction: without direct answer-label supervision, what signals can still guide the model toward reliable reasoning rather than merely self-consistent behavior?

Existing label-free approaches often construct rewards from the model's own generations, such as self-consistency, majority voting, confidence-based signals, or structural heuristics (Wang et al., 2022; Shafayat et al., 2025; Zhao et al., 2025; Prabhudesai et al., 2025; Xin et al., 2025). These signals can be effective when they are well aligned with correctness, but they remain internally defined by the model's own trajectory distribution. As observed in recent self-rewarding RL studies, such internal rewards may suffer from self-consistent illusion, reward hacking, or training collapse when the model repeatedly produces confident but incorrect trajectories (Shafayat et al., 2025; Zhang et al., 2025). This exposes a central limitation of purely internal pseudo-rewards: they measure agreement, confidence, or structure, but do not necessarily provide an external check on whether a trajectory reflects reliable reasoning. We therefore seek a label-free reward signal that is still externally verifiable, without using the main-task answer.

In this work, we identify such a phenomenon, which we call *Reasoning Warm-up*: when a model first solves a deterministic auxiliary task with an exact verifier before answering a more difficult reasoning problem, its success probability on the subsequent main task increases substantially. As shown in Table 1, trajectories with correct auxiliary solutions have consistently higher main-task success rates than trajectories with incorrect auxiliary solutions across different model backbones. Importantly, this effect is not merely a consequence of increasing prompt length or adding extra tokens. The verified auxiliary step appears to serve as an external anchor that encourages a more disciplined reasoning mode. When the auxiliary task is solved correctly, the subsequent main-task trajectory is more likely to remain structured, less likely to drift, and more likely to reach the correct final answer. This observation is related in spirit to decomposition-based prompting methods, such as least-to-most prompting and self-ask, which improve complex reasoning by explicitly solving simpler intermediate subproblems before producing the final answer (Zhou et al., 2022; Press et al.,

2023). However, our approach differs from these methods in that the auxiliary task is not used merely as an inference-time decomposition scaffold; instead, its verified outcome is later converted into a trajectory-level reward for policy optimization.

RWR uses a separately verifiable auxiliary task not to compute the main answer, but to provide a label-free reward signal for selecting full reasoning trajectories. A deterministic auxiliary task does not solve the main task by itself, nor does its verification result constitute full supervision for the main answer. Instead, auxiliary success provides a sparse but high-precision indication that a trajectory may originate from a more reliable latent reasoning state. Figure 1 provides a qualitative example of this phenomenon: under the same auxiliary-augmented prompt, a trajectory with a verified auxiliary answer also reaches the correct main-task answer, whereas a trajectory with an incorrect auxiliary answer subsequently follows an erroneous main-task reasoning path. Compared with purely internal criteria such as self-consistency, auxiliary verification offers an externally checkable event that is less dependent on the model's own agreement patterns and can therefore serve as a more reliable trajectory-level signal.

Table 1: Reasoning Warm-up statistics. Here $P_+ = P(S_M = 1 \mid S_A = 1)$, $P_- = P(S_M = 1 \mid S_A = 0)$.

Model	P_+	P_-	Δ
Qwen3-8B-Base	75.31%	46.40%	**28.91%**
Qwen2.5-7B	66.60%	45.39%	**21.21%**
Llama-3.2-3B-Instruct	39.89%	11.20%	**28.69%**

Motivated by Reasoning Warm-up, we propose *Reasoning Warm-up Reinforcement Learning* (RWR), a label-free reinforcement learning framework that incorporates verifiable auxiliary tasks into both generation and optimization. For each training instance, RWR constructs an auxiliary-augmented prompt consisting of a deterministic auxiliary task followed by the main mathematical reasoning problem. The model first solves the auxiliary task and then proceeds to the main task. During optimization, the reward is computed from two label-free components: a format reward that checks whether the trajectory is parseable, and an auxiliary-result reward that checks whether the auxiliary answer is correct. The main-task final-answer reward is never used during training. RWR uses the resulting hierarchical reward for GRPO-style policy optimization (Shao et al., 2024). In this way, auxiliary verification is converted into a training signal for reinforcement learning: it does not reveal the main-task answer, but biases policy updates toward trajectories that are more likely to support reliable downstream reasoning.

RWR is conceptually distinct from several neighboring paradigms. Unlike standard reinforcement learning from verifiable rewards, RWR does not optimize against the exact reward of the *main* task, even though such rewards may exist in principle for mathematics. Unlike process reward modeling, it does not require manually annotated intermediate reasoning steps or dense step-level supervision (Lightman et al., 2024). Unlike self-training methods such as STaR, which rely on final-answer correctness to select rationales (Zelikman et al., 2022), RWR intentionally withholds the main-task final-answer reward and instead uses auxiliary verification as an external proxy signal. Finally, unlike prompt scaffolding or curriculum learning, the auxiliary task is not introduced merely to simplify inference. Its verified outcome is explicitly incorporated into the reinforcement learning objective, allowing auxiliary success to influence which trajectories are reinforced during policy optimization.

We evaluate RWR on challenging mathematical reasoning benchmarks under a label-free training protocol. Across multiple settings, RWR demonstrates that auxiliary-task verification can serve as an effective optimization signal under the label-free training protocol. Rather than relying solely on internal pseudo-rewards such as self-consistency or structural heuristics, RWR provides a new route for unsupervised reinforcement learning by converting partial verifiability into a trajectory-level reward. Our contributions are summarized as follows:

1. We reveal and systematically study the empirical phenomenon of *Reasoning Warm-up*, showing that success on a deterministic auxiliary task is strongly associated with improved downstream reasoning performance.

2. We propose RWR, a label-free reinforcement learning framework that uses auxiliary-task verification as an externally checkable trajectory-level signal for policy optimization.

3. We conduct extensive experiments on multiple standard reasoning benchmarks, showing that RWR effectively improves models under a label-free training protocol and provides a promising step toward scalable self-improvement of large language models.

2 Related Work

2.1 Label-Free Reinforcement Learning for Reasoning

Reinforcement learning with verifiable rewards (RLVR) has become a prominent paradigm for eliciting reasoning abilities in large language models, achieving strong performance in mathematical and coding tasks (Shao et al., 2024; Guo et al., 2025; Luo et al., 2025). However, RLVR typically relies on high-quality ground-truth labels or task-specific verifiers, which limits its scalability in settings where direct supervision is unavailable or costly. To reduce this dependence, recent work has explored label-free or self-rewarding reinforcement learning. Representative approaches construct pseudo-rewards from the model's own generations, including self-consistency, majority voting, self-certainty, and entropy-based objectives (Wang et al., 2022; Shafayat et al., 2025; Zuo et al., 2025; Zhao et al., 2025; Prabhudesai et al., 2025). Another line of work investigates lightweight structural surrogate rewards, such as format and length signals, for mathematical reasoning without ground-truth answers (Xin et al., 2025). More recently, Co-rewarding introduces complementary supervision from data-side and model-side views to improve the stability of self-supervised RL (Zhang et al., 2025).

Despite their effectiveness, many label-free rewards are internally derived from the model's own output distribution and may become misaligned with correctness, especially in long-chain reasoning. RWR addresses this issue from a different angle: instead of relying solely on internal agreement or confidence, it uses the exact verification outcome of a deterministic auxiliary task as an externally checkable reward, while still withholding the main-task answer verifier during training.

2.2 Self-Feedback and Test-Time Adaptation

A related line of work improves LLM outputs through self-feedback or test-time adaptation. Self-Refine iteratively improves model generations by using feedback and refinement generated by the model itself (Madaan et al., 2023), while Reflexion equips language agents with verbal feedback and memory to improve future decisions (Shinn et al., 2023). Test-Time Reinforcement Learning further adapts models on unlabeled test problems using intrinsic verifiers such as majority voting or confidence-based signals (Zuo et al., 2025; Prabhudesai et al., 2025; Zhao et al., 2025). These methods show that feedback generated without human labels can improve model behavior, but their learning signals are often internal and may be unreliable when the model repeatedly produces consistent but incorrect reasoning paths. Process-level verification provides another way to improve reasoning reliability. Step-level supervision can offer stronger guidance than final-answer feedback alone (Lightman et al., 2024). Recent work such as Math-Shepherd reduces the need for human annotations by automatically constructing process-level supervision and using it to verify and reinforce intermediate reasoning steps (Wang et al., 2024a). In contrast, RWR does not require step-level verification of the main-task solution. It verifies a separately constructed auxiliary task and uses its outcome as a surrogate reward for the full auxiliary-main trajectory.

3 Method

We propose *Reasoning Warm-up Reinforcement Learning* (RWR), a label-free reinforcement learning framework for mathematical reasoning. During optimization, RWR never uses the ground-truth final answer of the main problem as a reward. Instead, it prepends a deterministic and exactly verifiable auxiliary task to the main problem, and converts the auxiliary verification result into a trajectory-level reward for the full auxiliary-main response. This design follows from the Reasoning Warm-up observation in Table 1: auxiliary-task success is positively associated with subsequent main-task success. Thus, the auxiliary task is not trained as a separate objective; it serves as an externally verifiable signal for selecting and reinforcing more reliable reasoning trajectories.

3.1 Auxiliary Task Construction

For each main problem x, RWR constructs a deterministic auxiliary task $a_x = \mathcal{A}(x)$ that is independent of the main-task answer. The auxiliary task is not intended to solve or simplify the main problem; instead, it provides a verifiable warm-up event before the main reasoning stage. In our implementation, a_x is generated programmatically using simple deterministic operations, such as memory-based counting, sorting, or arithmetic manipulation. Since the auxiliary input and operation are fully specified, its answer $y_A^\star(a_x)$ can be computed exactly, and the auxiliary verifier is

$$V_A(a_x, \hat{y}_A) = \mathbf{1}\left[\hat{y}_A = y_A^\star(a_x)\right], \tag{1}$$

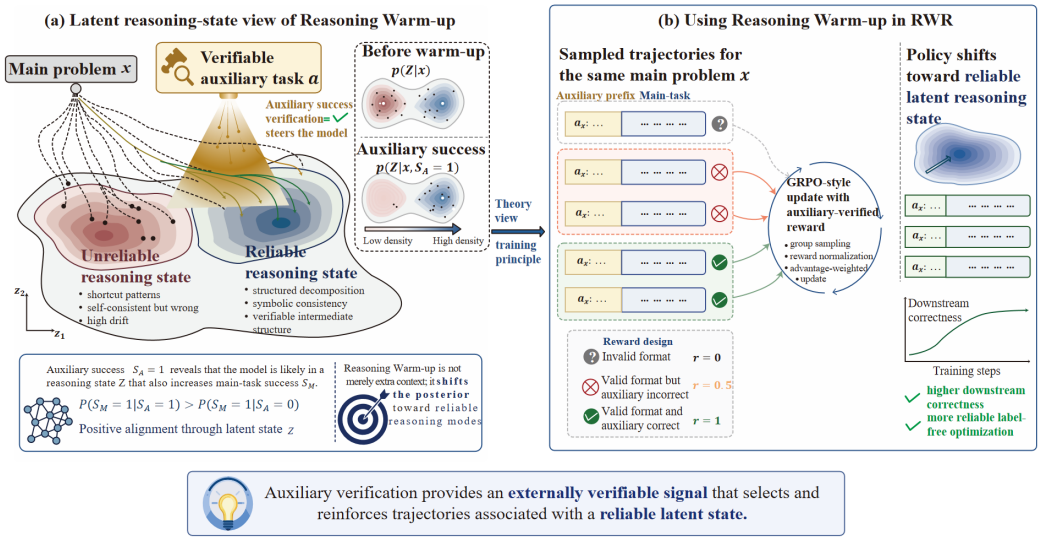

Figure 2: Illustration of RWR. Left: auxiliary-task success selects trajectories associated with reliable latent reasoning states, and right: RWR converts this verification signal into a trajectory-level reward for GRPO optimization.

where $\hat{y}_A$ is extracted from the model output. This ensures that RWR obtains an exact training signal without using the main-task ground-truth answer.

3.2 Auxiliary-Augmented Rollout

Let $x \sim \mathcal{D}$ be a main mathematical reasoning problem and let π_θ be the policy model. For each x, RWR constructs a deterministic auxiliary task $a_x = \mathcal{A}(x)$. The training prompt contains two parts: the auxiliary task followed by the main problem. The model then generates a two-stage trajectory

$$\tau = (u, v), \tag{2}$$

where u is the auxiliary-task response and v is the main-task response. The trajectory distribution factorizes as $\pi_\theta(\tau \mid x) = \pi_\theta(u \mid p_A(x))\pi_\theta(v \mid p_M(x, u))$, where $p_A(x)$ denotes the auxiliary-task prompt and $p_M(x, u)$ denotes the main-task prompt conditioned on the auxiliary response.

This construction is important because the main-task response v is generated after the auxiliary response u. Thus, the auxiliary task is not an independent side objective. It forms a verifiable prefix of the same reasoning trajectory. If the auxiliary response is correct, the entire trajectory is more likely to correspond to a reliable reasoning process. RWR exploits this property by assigning the auxiliary-based reward to the full trajectory (u, v).

3.3 Trajectory-Level Warm-up Reward

For a sampled trajectory $\tau_i = (u_i, v_i)$, RWR computes a scalar reward from two observable signals: format validity and auxiliary correctness. The format verifier V_{fmt} checks whether the model output follows the required structure, such as producing a valid auxiliary answer field and a valid main-answer field. We define $R_i^{\text{fmt}} = V_{\text{fmt}}(\tau_i) \in \{0, 1\}$.

If the output is well-formed, we extract the auxiliary answer using $E_A(u_i)$ and evaluate it with the exact auxiliary verifier: $R_i^{\text{aux}} = V_A(a_x, E_A(u_i)) \in \{0, 1\}$.

The main-task answer is not checked during training.

The final RWR reward is

$$R_i^{\text{RWR}} = R_i^{\text{fmt}} \left(\frac{1}{2} + \frac{1}{2} R_i^{\text{aux}} \right). \tag{3}$$

Equivalently,

$$R_i^{\mathrm{RWR}} = \begin{cases} 0, & R_i^{\mathrm{fmt}} = 0, \\ \frac{1}{2}, & R_i^{\mathrm{fmt}} = 1 \text{ and } R_i^{\mathrm{aux}} = 0, \\ 1, & R_i^{\mathrm{fmt}} = 1 \text{ and } R_i^{\mathrm{aux}} = 1. \end{cases} \tag{4}$$

This reward has a simple interpretation. Invalid outputs receive zero reward because they cannot be reliably parsed. Valid outputs receive partial reward because they satisfy the structural requirements for optimization. Valid outputs with correct auxiliary answers receive the maximum reward because the verified auxiliary success indicates a higher likelihood of correct downstream reasoning. The same scalar reward R_i^{RWR} is assigned to the whole trajectory $\tau_i = (u_i, v_i)$, so the policy update affects both the auxiliary response and the main-task continuation.

3.4 RWR Optimization

RWR uses a GRPO-style objective. For each problem x, we sample a group of G trajectories from the old policy:

$$\mathcal{G}(x) = \{\tau_i = (u_i, v_i)\}_{i=1}^{G}, \qquad \tau_i \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x). \tag{5}$$

Each trajectory receives the reward R_i^{RWR}. The group mean and standard deviation are

$$\mu_{\mathrm{RWR}} = \frac{1}{G}\sum_{j=1}^{G} R_j^{\mathrm{RWR}}, \qquad \sigma_{\mathrm{RWR}} = \sqrt{\frac{1}{G}\sum_{j=1}^{G}\left(R_j^{\mathrm{RWR}} - \mu_{\mathrm{RWR}}\right)^2}. \tag{6}$$

The group-relative advantage is

$$\widehat{A}_i^{\mathrm{RWR}} = \frac{R_i^{\mathrm{RWR}} - \mu_{\mathrm{RWR}}}{\sigma_{\mathrm{RWR}} + \epsilon}. \tag{7}$$

Let $\tau_i = (o_{i,1}, \ldots, o_{i,T_i})$ be the full token sequence of the generated trajectory. For each token, define

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid x, o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid x, o_{i,<t})}. \tag{8}$$

The RWR objective is

$$\begin{aligned} \mathcal{J}_{\mathrm{RWR}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\left(\rho_{i,t}(\theta)\widehat{A}_i^{\mathrm{RWR}}, \mathrm{clip}\left(\rho_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\widehat{A}_i^{\mathrm{RWR}}\right) \\ - \beta\mathcal{D}_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\mathrm{ref}}\right). \end{aligned} \tag{9}$$

The training loss is

$$\mathcal{L}_{\mathrm{RWR}}(\theta) = -\mathcal{J}_{\mathrm{RWR}}(\theta). \tag{10}$$

The key point is that $\widehat{A}_i^{\mathrm{RWR}}$ is applied to all tokens in the full trajectory, not only to the auxiliary answer. The trajectory-level advantage weights all tokens in the response, so both the auxiliary prefix and the main-task continuation contribute to the policy objective. In this way, RWR uses the auxiliary task reward as a shared trajectory reward for the combined auxiliary-main response, enabling policy optimization without using the main-task ground-truth answer. The full training procedure is provided in Algorithm 1 and the overall pipeline is illustrated in Figure 2.

4 Theoretical Justification of Reasoning Warm-up

We provide a concise theoretical explanation for why auxiliary-task verification can serve as a useful trajectory-level signal for RWR. Let $S_A \in \{0, 1\}$ denote whether the auxiliary task is solved correctly, and let $S_M \in \{0, 1\}$ denote whether the main task is solved correctly. During training, S_A is observable through the auxiliary verifier, while S_M is withheld from reward computation.

We introduce a latent reasoning state Z, which captures the underlying reasoning regime of a generated trajectory. Define

$$a(z) = \Pr(S_A = 1 \mid Z = z), \qquad m(z) = \Pr(S_M = 1 \mid Z = z). \tag{11}$$

Let $r_A = \Pr(S_A = 1)$, $r_M = \Pr(S_M = 1)$, and define the auxiliary–main reasoning alignment as

$$\Gamma = \mathrm{Cov}_Z\big(a(Z), m(Z)\big). \tag{12}$$

A positive Γ means that latent states that make the auxiliary task more likely to succeed also make the main task more likely to succeed.

Assumption 4.1 (Latent reasoning sufficiency)**.** *Conditioned on the latent reasoning state Z, the residual randomness in auxiliary-task success and main-task success is independent:*

$$\Pr(S_A = 1, S_M = 1 \mid Z) = \Pr(S_A = 1 \mid Z)\Pr(S_M = 1 \mid Z). \tag{13}$$

Theorem 4.2 (Verifier-induced Reasoning Warm-up)**.** *Suppose Assumption 4.1 holds and $0 < r_A < 1$. Then*

$$\Pr(S_M = 1 \mid S_A = 1) = r_M + \frac{\Gamma}{r_A}, \tag{14}$$

$$\Pr(S_M = 1 \mid S_A = 0) = r_M - \frac{\Gamma}{1 - r_A}. \tag{15}$$

Therefore, the conditional success lift induced by auxiliary verification is

$$\Pr(S_M = 1 \mid S_A = 1) - \Pr(S_M = 1 \mid S_A = 0) = \frac{\Gamma}{r_A(1 - r_A)}. \tag{16}$$

Consequently, if $\Gamma > 0$, then

$$\Pr(S_M = 1 \mid S_A = 1) > \Pr(S_M = 1 \mid S_A = 0). \tag{17}$$

Theorem 4.2 formalizes the empirical Reasoning Warm-up phenomenon. It shows that auxiliary-task success increases the posterior probability of main-task success whenever auxiliary success is positively aligned with the latent reasoning state governing downstream correctness. This directly supports the design of RWR: the auxiliary verifier does not need to reveal the main-task answer; it only needs to provide an externally checkable signal that selects trajectories more likely to originate from reliable reasoning states. RWR then uses this signal as a trajectory-level reward for policy optimization. Full proofs and additional implications are provided in Appendix A.7.

5 Experiments

5.1 Experiment Setup

Backbone Models and Baselines.

Following (Zhang et al., 2025), we validate the effectiveness of RWR across a series of models, including the Qwen2.5 series (Qwen2.5-7B (Yang et al., 2024)), the Qwen3 series (Qwen3-8B (Yang et al., 2025)), and the Llama series (Llama-3.2-3B-Instruct (Grattafiori et al., 2024)). We establish the following methods as our baselines for comparison: Self-Certainty (Zhao et al., 2025), Entropy (Prabhudesai et al., 2025), Majority Voting (Shafayat et al., 2025), and Co-rewarding (Zhang et al., 2025). For reference, we also report results obtained by training with ground-truth final-answer rewards, denoted as GT-Reward, to provide a comprehensive comparison.

Implementation Details. Following Co-Rewarding, we use the MATH dataset (Hendrycks et al., 2021) and DAPO-14k (Yu et al., 2025) as training sets, and evaluate the model on several evaluation benchmarks, including MATH500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), AMC (Li et al., 2024), AIME24, IFEval (Zhou et al., 2023) and MMLU-Pro (Wang et al., 2024b). During RL training, the batch size is set to 128, the number of sampled rollouts per prompt is set to 8, and the sampling temperature is set to 1.0. The model is optimized using AdamW with a learning rate of 2×10^{-6}. At test time, we report pass@1 accuracy. More detailed training and evaluation settings are provided in Appendix A.6.

5.2 Main Results

As shown in Table 2, we report the results of models trained on MATH under the main-answer-withheld setting. RWR improves model performance under label-free optimization and achieves

Table 2: Comparison of methods trained on the MATH dataset. Values are percentages. **Bold** and underlined numbers indicate the best and second-best distinct scores among label-free methods for each backbone; ties share the same rank. Before RL and GT-Reward are reference rows and are not ranked. For Co-rewarding, we follow the original implementation and use an additional **Qwen3-32B** model to rephrase mathematical questions.

Methods	**MATH500**	**GSM8K**	**AMC**	**AIME24**	**IFEval**	**MMLU-Pro**
			Qwen3-8B-Base			
Before RL	72.40	27.82	20.93	3.75	50.89	52.92
GT-Reward	82.60	87.26	54.22	17.15	52.78	57.11
Self-Certainty	80.20	80.74	50.75	15.73	50.98	54.17
Entropy	80.20	87.19	49.54	15.63	51.81	54.86
Majority-Voting	79.80	89.76	49.09	15.83	51.80	56.93
Co-rewarding	**80.80**	92.42	53.46	14.48	**60.70**	**57.95**
RWR	**80.80**	**93.18**	**55.42**	**16.67**	55.82	53.14
			Qwen2.5-7B			
Before RL	60.20	24.71	15.81	2.81	38.19	44.76
GT-Reward	67.80	88.02	45.63	14.06	41.49	41.12
Self-Certainty	64.40	84.31	38.55	8.75	37.24	43.30
Entropy	63.20	81.43	39.61	10.73	40.33	42.79
Majority-Voting	65.80	84.53	40.96	11.04	38.60	**43.35**
Co-rewarding	64.20	**89.31**	**42.77**	**11.98**	41.82	37.45
RWR	**66.60**	88.66	38.55	6.67	**42.14**	40.62
			Llama-3.2-3B-Instruct			
Before RL	39.20	65.73	10.54	3.75	57.32	31.14
GT-Reward	47.00	77.94	22.14	11.67	47.51	34.32
Self-Certainty	43.40	74.91	18.83	6.88	**54.88**	33.34
Entropy	43.40	66.19	20.18	6.56	54.70	33.52
Majority-Voting	46.80	78.77	20.48	9.27	47.96	33.18
Co-rewarding	**49.80**	**79.30**	**22.59**	**10.73**	49.90	**33.61**
RWR	48.20	79.23	20.48	10.00	51.39	31.92

competitive results compared with existing label-free baselines such as Self-Certainty, Entropy, and Majority-Voting across different model backbones and mathematical reasoning benchmarks. These results demonstrate that auxiliary-task verification provides an effective signal for label-free policy optimization. Table 3 further reports results when training on DAPO-14k. RWR remains competitive across models and datasets, indicating that the benefit of Reasoning Warm-up is not tied to a specific training set. Compared with GT-Reward, RWR does not use the main-task answer verifier during training, yet it substantially narrows the gap between label-free optimization and verifiable-reward training. Overall, the results suggest that using auxiliary-task verification as a trajectory-level reward can improve mathematical reasoning under label-free training, while maintaining reasonable generalization to instruction-following and multi-task evaluations.

5.3 Ablation Study

In this section, we conduct ablation experiments. Unless otherwise specified, all ablation experiments are conducted on Qwen3-8B-Base with MATH as the training dataset.

Reward design. Table 4 reports the ablation results of the RWR reward. The full reward combines format validity and auxiliary-task correctness, corresponding to the reward used in our main

Table 3: Comparison of methods trained on DAPO-14k. Values are percentages. **Bold** and underlined numbers indicate the best and second-best distinct scores among label-free methods for each backbone; ties share the same rank. Before RL and GT-Reward are reference rows and are not ranked.

Methods	**MATH500**	**GSM8K**	**AMC**	**AIME24**	**IFEval**	**MMLU-Pro**
Qwen3-8B-Base						
Before RL	72.40	27.82	20.93	3.75	50.89	52.92
GT-Reward	86.60	87.19	61.75	24.58	53.11	60.27
Self-Certainty	**82.00**	77.63	49.85	15.00	50.58	54.24
Entropy	79.40	80.82	45.48	15.00	51.56	54.57
Majority-Voting	78.60	91.66	50.00	11.25	51.54	55.65
Co-rewarding	80.60	**94.01**	**54.37**	**16.35**	**53.31**	**59.83**
RWR	79.60	91.30	49.40	13.33	51.76	54.54
Llama-3.2-3B-Instruct						
Before RL	39.20	65.73	10.54	3.75	57.32	31.14
GT-Reward	49.40	78.17	25.90	9.17	53.10	33.83
Self-Certainty	42.40	74.71	17.32	4.79	54.50	33.51
Entropy	44.00	65.85	17.32	6.56	**55.78**	31.95
Majority-Voting	42.80	70.96	17.62	**8.74**	54.07	32.95
Co-rewarding	**49.80**	78.62	19.73	8.02	51.92	**34.46**
RWR	**49.80**	**78.77**	**21.69**	3.33	52.13	32.08

Table 4: Ablation study on reward design.

Variant	MATH500	GSM8K	AMC	AIME24
Full reward	80.80	93.18	55.42	16.67
w/o aux. reward	62.80	86.66	45.78	6.67
w/o format reward	75.60	91.58	54.22	10.00

Table 5: Ablation study on auxiliary-task design.

Auxiliary task	MATH500	GSM8K	AMC	AIME24
Memory task	80.80	93.18	55.42	16.67
Sum number	73.80	90.83	46.99	13.33
Max number	74.40	90.60	49.40	10.00

experiments. Removing the auxiliary-result reward reduces the objective to format-only optimization, which can encourage parseable outputs but lacks a signal for distinguishing whether the trajectory passes the verifiable warm-up stage. Removing the format reward, on the other hand, makes auxiliary verification less reliable because malformed outputs may lead to unstable answer extraction. The comparison shows that the full reward provides the most effective training signal, suggesting that format validity and auxiliary correctness are complementary: the former ensures reliable parsing, while the latter supplies the key trajectory-level verification signal.

Auxiliary-task design. Table 5 compares three deterministic auxiliary-task designs. The first is a memory-based task, where the model is given a short sequence of simple numbers and is required to output the number at a specified position, or perform a simple arithmetic operation on numbers at specified positions. The second is a number-sum task, where the model is asked to compute the sum of all numbers appearing in the main problem. The third is a maximum-number task, where the model is asked to identify the largest number in the main problem. For all three tasks, the auxiliary ground-truth answer can be generated automatically by a program, without any manual annotation or access to the main-task answer.

The results show that the memory-based task provides the strongest auxiliary signal in our setting. This is likely because it activates basic memory, attention, and arithmetic manipulation abilities, which are also important components of mathematical reasoning. By contrast, the number-sum and maximum-number tasks appear closely related to the main problem because they require the model to focus on numerical information in the question. However, numbers in mathematical problems can be semantically noisy: some appear in exponents, denominators, radicals, indices, or other structured expressions. As a result, the model's interpretation of the relevant numbers may deviate

from the automatically generated auxiliary ground truth, making these auxiliary signals less stable. These results suggest that an effective auxiliary task should be exactly verifiable while also eliciting reasoning-relevant capabilities in a robust and unambiguous way.

5.4 Policy Learning and Inference-Time Warm-up

To examine the contributions of policy learning and inference-time warm-up, we evaluate Qwen3-8B-Base and the RWR-trained policy with main-only and verified warm-up prompts (Table 6). RWR improves all four mathematical benchmarks under main-only prompting. Applying warm-up to the base model yields smaller, mixed changes, whereas it further improves the RWR-trained policy on all four mathematical benchmarks. For IFEval, main-only prompting performs better for both policies, illustrating task-dependent effects of the composite prompt. Appendix A.10 reports the corresponding comparisons across all trained baselines.

Table 6: Separating RWR training and inference-time warm-up on Qwen3-8B-Base. Warm-up uses the verified auxiliary prompt. Values are percentages; IFEval uses strict evaluation.

RWR training	Inference prompt	MATH500	GSM8K	AMC	AIME24	IFEval	MMLU-Pro
No	Main-only	72.40	27.82	20.93	3.75	50.89	52.92
No	Warm-up	73.60	28.81	20.48	6.67	47.13	48.92
Yes	Main-only	76.40	91.28	50.60	10.00	58.78	52.79
Yes	Warm-up	80.80	93.18	55.42	16.67	55.82	53.14

6 Conclusion

We presented RWR, a label-free reinforcement learning framework that improves reasoning without using the main-task final-answer reward during optimization. Motivated by the Reasoning Warm-up phenomenon, RWR converts the exact verification outcome of a deterministic auxiliary task into a trajectory-level reward for GRPO-style policy optimization. Experiments on multiple reasoning benchmarks show that verifiable auxiliary signals can effectively guide label-free optimization and provide a practical alternative to purely internal pseudo-rewards such as self-consistency or confidence-based signals. These findings suggest a promising direction for scalable self-improvement: designing auxiliary tasks whose verifiable outcomes reveal useful information about the reliability of model-generated reasoning trajectories.

Acknowledgments and Disclosure of Funding

JN and BH were supported by RGC Research Fellow Scheme No. RFS2627-2S02, NSFC Major Research Plan No. 92570109, RGC Young Collaborative Research Grant No. C2005-24Y, RGC General Research Fund No. 12202026, RGC General Research Fund No. 12200725, NSFC General Program No. 62376235, TCL Collaborative Research Fund, Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme, HKBU Faculty Niche Research Areas No. RC-FNRA-IG/25-26/SCI/05, and HKBU CSD Departmental Incentive Scheme. JN and XMT were supported in part by the Natural Science Foundation of China under Grant 62571507. MKN is supported in part by the Department of Science and Technology of Guangdong Province (GDSTC): Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme, the National Key Research and Development Program of China (Grant No. 2024YFE0202900), and Research Grants Council (RGC) of Hong Kong under the General Research Fund (GRF) (Grant No. 12300125).

Competing interests. The authors declare no other competing financial interests.

References

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. (2022). Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems, NeurIPS*.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Huang, J., Gu, S., Hou, L., Wu, Y., Wang, X., Yu, H., and Han, J. (2023). Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing, EMNLP*.

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Hall, E., Carbune, V., Rastogi, A., et al. (2023). Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S., Rasul, K., Yu, L., Jiang, A. Q., Shen, Z., et al. (2024). Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. (2024). Let's verify step by step. In *The twelfth international conference on learning representations*.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Zhang, T., Li, L. E., Popa, R. A., and Stoica, I. (2025). Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. (2023). Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems, NeurIPS*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems, NeurIPS*.

Prabhudesai, M., Chen, L., Ippoliti, A., Fragkiadaki, K., Liu, H., and Pathak, D. (2025). Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*.

Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., and Lewis, M. (2023). Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shafayat, S., Tajwar, F., Salakhutdinov, R., Schneider, J., and Zanette, A. (2025). Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems, NeurIPS*.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. (2024a). Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. (2024b). Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems, NeurIPS*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems, NeurIPS*.

Xin, R., Liu, H., Wang, Z., Zhang, Y., Sui, D., Hu, X., and Wang, B. (2025). Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *arXiv preprint arXiv:2505.19439*.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. (2024). Qwen2.5 technical report. *CoRR*.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. (2025). Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., and Weston, J. (2024). Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. (2022). Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems, NeurIPS*.

Zhang, Z., Zhu, J., Ge, X., Zhao, Z., Zhou, Z., Li, X., Feng, X., Yao, J., and Han, B. (2025). Co-rewarding: Stable self-supervised rl for eliciting reasoning in large language models. *arXiv preprint arXiv:2508.00410*.

Zhao, X., Kang, Z., Feng, A., Levine, S., and Song, D. (2025). Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al. (2022). Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. (2023). Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Zuo, Y., Zhang, K., Sheng, L., Qu, S., Cui, G., Zhu, X., Li, H., Zhang, Y., Long, X., Hua, E., et al. (2025). Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*.

A Appendix

A.1 Limitations

RWR provides a label-free optimization framework by using verifiable auxiliary tasks as trajectory-level reward signals. Nevertheless, the current study has several limitations.

- **Dependence on auxiliary–main alignment.** The effectiveness of RWR depends on whether auxiliary-task success is positively associated with main-task reasoning quality. If the auxiliary task is too simple, too difficult, or weakly related to the reasoning ability required by the main task, its verification outcome may provide a less informative optimization signal.
- **Scope of auxiliary tasks.** The current auxiliary tasks are mainly deterministic symbolic or memory-based tasks, such as counting, sorting, and simple arithmetic manipulation. Although these tasks are exactly verifiable, they may not cover all reasoning skills required in broader domains.

A.2 Social Impact

RWR studies label-free reinforcement learning for improving reasoning models through verifiable auxiliary signals. Its potential impacts include both benefits and risks.

- **Positive impact.** RWR reduces the dependence on main-task answer labels during optimization, which may lower annotation costs and support more scalable reasoning-oriented post-training. By using deterministic auxiliary verifiers, it also provides a more transparent reward source than purely internal pseudo-rewards.
- **Potential misuse.** Improvements in reasoning ability may increase model capability in both beneficial and harmful applications. Therefore, RWR-trained models should be deployed with appropriate safeguards, especially in high-stakes domains such as medicine, law, finance, and education.

A.3 Algorithm of RWR

Algorithm 1 summarizes the training procedure of RWR. For each main problem, RWR first constructs a deterministic auxiliary task and samples a group of auxiliary-main trajectories. Each trajectory is rewarded only by format validity and auxiliary-task correctness; the ground-truth answer of the main task is never used for reward computation. The resulting scalar reward is assigned to the full trajectory and optimized with a GRPO-style objective.

A.4 Attention Visualization for Auxiliary-Task Reasoning

We provide a qualitative analysis of how the model attends to the auxiliary-task input under different sampled continuations. The analysis is based on a representative dual-task example consisting of a memory-based auxiliary task and a mathematical reasoning problem. In the auxiliary task, the model is given the sequence

$$820,\ 937,\ 820,\ 515,\ 123,\ 755,\ 783,\ 353,\ 544 \tag{18}$$

and is asked to output the 6th number from the end. The correct auxiliary answer is 515. Under the same prompt, one sampled continuation produces the correct auxiliary answer, `<aux>515</aux>`, and also solves the mathematical problem correctly; another sampled continuation produces the incorrect auxiliary answer, `<aux>755</aux>`, and fails on the mathematical problem.

To visualize the attention pattern, we use teacher forcing on each sampled continuation and extract the attention assigned to the original input prompt when the model processes the generated auxiliary answer. For the successful continuation, the query tokens correspond to the generated answer "515"; for the failed continuation, the query tokens correspond to the generated answer "755". We average attention scores over all heads and selected upper layers, and aggregate the resulting token-level scores back to the original prompt text. Thus, the visualization is conditioned on the same input prompt but different sampled continuations.

Algorithm 1 Training Procedure of Reasoning Warm-up Reinforcement Learning

Require: Training distribution $\mathcal{D}$; policy π_θ; reference policy π_{ref}; auxiliary constructor $\mathcal{A}$; format verifier V_{fmt}; auxiliary verifier V_A; auxiliary answer extractor E_A; group size G

for each training iteration **do**
  Set the old policy $\pi_{\theta_{\mathrm{old}}} \leftarrow \pi_\theta$
  Sample a main problem $x \sim \mathcal{D}$
  Construct the auxiliary task $a_x = \mathcal{A}(x)$
  Construct the auxiliary-augmented prompt $p_{\mathrm{RWR}}(x, a_x)$
  for $i = 1, \ldots, G$ **do**
    Sample a full trajectory $\tau_i = (u_i, v_i) \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid p_{\mathrm{RWR}}(x, a_x))$
    Compute format reward $R_i^{\mathrm{fmt}} = V_{\mathrm{fmt}}(\tau_i)$
    if $R_i^{\mathrm{fmt}} = 1$ **then**
      Extract auxiliary answer $\hat{y}_{A,i} = E_A(u_i)$
      Compute auxiliary reward $R_i^{\mathrm{aux}} = V_A(a_x, \hat{y}_{A,i})$
    else
      Set $R_i^{\mathrm{aux}} = 0$
    end if
    Compute trajectory reward
    $$R_i^{\mathrm{RWR}} = R_i^{\mathrm{fmt}} \left(\frac{1}{2} + \frac{1}{2} R_i^{\mathrm{aux}} \right)$$
  end for
  Compute μ_{RWR} and σ_{RWR} from $\{R_i^{\mathrm{RWR}}\}_{i=1}^{G}$
  Compute group-relative advantages for all $i = 1, \ldots, G$
  $$\widehat{A}_i^{\mathrm{RWR}} = \frac{R_i^{\mathrm{RWR}} - \mu_{\mathrm{RWR}}}{\sigma_{\mathrm{RWR}} + \epsilon}$$
  Update π_θ by minimizing $\mathcal{L}_{\mathrm{RWR}}(\theta)$
end for

Figure 3 shows the attention distribution over the full original prompt. Darker colors indicate higher attention for the corresponding continuation. The difference panel visualizes the signed difference between the successful and failed continuations, highlighting regions of the prompt that receive relatively higher attention in one continuation than in the other. Figure 4 zooms in on the numerical sequence in the auxiliary task.

These visualizations provide a fine-grained explanation for why auxiliary-task success can be predictive of main-task success. The auxiliary task is solved before the main problem within the same generated trajectory. Therefore, the attention pattern during auxiliary-answer generation reflects the model's early prompt-grounding and working-memory state before it enters the main reasoning stage. In the successful continuation, the model attends more consistently to the task-relevant auxiliary information and outputs the correct auxiliary answer. In the failed continuation, the model places stronger attention on the distractor number it later outputs, indicating an unreliable grounding of the auxiliary input. This early misalignment is followed by an incorrect main-task reasoning path in the same trajectory. Thus, the visualization supports the interpretation that auxiliary verification captures trajectory-level reasoning reliability, rather than merely checking whether the auxiliary output format is correct.

A.5 Auxiliary Task Details

We use deterministic auxiliary tasks with exact programmatic verification. Each auxiliary task consists of an input object, an operation, and a required output format. The auxiliary answer is computed by a deterministic script and is independent of the main-task answer.

Counting from the end. Given a sequence of integers, the model is asked to count from the end and output the k-th number.

Sorting and selection. Given a sequence of integers, the model is asked to sort the sequence in increasing order and output the k-th element.

Arithmetic manipulation. Given a set of integers and a deterministic arithmetic instruction, the model is asked to compute the specified result.

All auxiliary answers are verified by exact string extraction and numerical comparison. The required auxiliary format is `<aux>...</aux>`.

A.6 Implementation Details

We implement RWR based on the verl framework. Unless otherwise specified, all RL experiments use the same optimization configuration across methods for fair comparison. During training, the global batch size is set to 128, and each prompt is sampled with 8 rollouts. The rollout sampling temperature is set to 1.0. We optimize the policy using AdamW with a learning rate of 2×10^{-6}. The maximum prompt length is set to 1024, and the maximum response length is set to 4096. The KL coefficient is set to 0.05, and the clipping ratio in GRPO is set to 0.2. Each model is trained for 150 optimization steps.

At test time, we use a sampling temperature of 0.2. For mathematical reasoning benchmarks, we report pass@1 accuracy. The final answer is extracted from the last valid $\backslash boxed\{\cdot\}$ expression and compared with the reference answer after normalization. For IFEval and MMLU-Pro, we follow the official evaluation protocols to assess instruction-following and general multi-task performance. All experiments are conducted on 4 NVIDIA H20 GPUs.

A.7 Proofs for Reasoning Warm-up

This appendix provides the proofs for Section 4. We first prove Theorem 4.2, then discuss verifier-based trajectory reweighting and the identical-task limiting case.

A.7.1 Proof of Theorem 4.2

Recall that

$$a(z) = \Pr(S_A = 1 \mid Z = z), \qquad m(z) = \Pr(S_M = 1 \mid Z = z), \tag{19}$$

and that

$$r_A = \Pr(S_A = 1) = \mathbb{E}_Z[a(Z)], \qquad r_M = \Pr(S_M = 1) = \mathbb{E}_Z[m(Z)]. \tag{20}$$

Since $0 < r_A < 1$, both conditional probabilities $\Pr(S_M = 1 \mid S_A = 1)$ and $\Pr(S_M = 1 \mid S_A = 0)$ are well-defined.

We first compute $\Pr(S_M = 1 \mid S_A = 1)$. By conditional probability,

$$\Pr(S_M = 1 \mid S_A = 1) = \frac{\Pr(S_M = 1, S_A = 1)}{\Pr(S_A = 1)}. \tag{21}$$

Using the law of total expectation,

$$\Pr(S_M = 1, S_A = 1) = \mathbb{E}_Z\left[\Pr(S_M = 1, S_A = 1 \mid Z)\right]. \tag{22}$$

By Assumption 4.1,

$$\begin{aligned} \Pr(S_M = 1, S_A = 1 \mid Z = z) &= \Pr(S_M = 1 \mid Z = z) \Pr(S_A = 1 \mid Z = z) \\ &= m(z)a(z). \end{aligned} \tag{23}$$

Therefore,

$$\Pr(S_M = 1, S_A = 1) = \mathbb{E}_Z[m(Z)a(Z)]. \tag{24}$$

By the covariance identity,

$$\mathbb{E}_Z[m(Z)a(Z)] = \mathbb{E}_Z[m(Z)]\mathbb{E}_Z[a(Z)] + \mathrm{Cov}_Z(a(Z), m(Z)). \tag{25}$$

Using $\mathbb{E}_Z[m(Z)] = r_M$, $\mathbb{E}_Z[a(Z)] = r_A$, and $\Gamma = \mathrm{Cov}_Z(a(Z), m(Z))$, we obtain

$$\Pr(S_M = 1, S_A = 1) = r_M r_A + \Gamma. \tag{26}$$

Thus,

$$\begin{aligned} \Pr(S_M = 1 \mid S_A = 1) &= \frac{r_M r_A + \Gamma}{r_A} \\ &= r_M + \frac{\Gamma}{r_A}. \end{aligned} \tag{27}$$

We next compute $\Pr(S_M = 1 \mid S_A = 0)$. Since

$$\Pr(S_M = 1) = \Pr(S_M = 1, S_A = 1) + \Pr(S_M = 1, S_A = 0), \tag{28}$$

we have

$$\begin{aligned} \Pr(S_M = 1, S_A = 0) &= r_M - (r_M r_A + \Gamma) \\ &= r_M(1 - r_A) - \Gamma. \end{aligned} \tag{29}$$

Therefore,

$$\begin{aligned} \Pr(S_M = 1 \mid S_A = 0) &= \frac{\Pr(S_M = 1, S_A = 0)}{\Pr(S_A = 0)} \\ &= \frac{r_M(1 - r_A) - \Gamma}{1 - r_A} \\ &= r_M - \frac{\Gamma}{1 - r_A}. \end{aligned} \tag{30}$$

Combining Eq. (27) and Eq. (30), we obtain

$$\begin{aligned} &\Pr(S_M = 1 \mid S_A = 1) - \Pr(S_M = 1 \mid S_A = 0) \\ &= \left(r_M + \frac{\Gamma}{r_A}\right) - \left(r_M - \frac{\Gamma}{1 - r_A}\right) \\ &= \frac{\Gamma}{r_A} + \frac{\Gamma}{1 - r_A} = \frac{\Gamma}{r_A(1 - r_A)}. \end{aligned} \tag{31}$$

If $\Gamma > 0$, then

$$\Pr(S_M = 1 \mid S_A = 1) > \Pr(S_M = 1 \mid S_A = 0). \tag{32}$$

This completes the proof.

A.7.2 Verifier-Based Reweighting

Theorem 4.2 also justifies verifier-based trajectory reweighting. For $\lambda \geq 0$, define

$$w_\lambda(S_A) = 1 + \lambda S_A. \tag{33}$$

Let the main-task success rate under the reweighted trajectory distribution be

$$r_M^{(\lambda)} = \frac{\mathbb{E}[w_\lambda(S_A) S_M]}{\mathbb{E}[w_\lambda(S_A)]}. \tag{34}$$

The denominator is

$$\mathbb{E}[w_\lambda(S_A)] = 1 + \lambda r_A. \tag{35}$$

The numerator is

$$\begin{aligned} \mathbb{E}[w_\lambda(S_A) S_M] &= \mathbb{E}[(1 + \lambda S_A) S_M] \\ &= \mathbb{E}[S_M] + \lambda \mathbb{E}[S_A S_M]. \end{aligned} \tag{36}$$

From the proof above,

$$\mathbb{E}[S_A S_M] = \Pr(S_A = 1, S_M = 1) = r_A r_M + \Gamma. \tag{37}$$

Therefore,

$$\begin{aligned} r_M^{(\lambda)} &= \frac{r_M + \lambda(r_A r_M + \Gamma)}{1 + \lambda r_A} \\ &= r_M + \frac{\lambda \Gamma}{1 + \lambda r_A}. \end{aligned} \tag{38}$$

Thus, if $\Gamma > 0$ and $\lambda > 0$, then

$$r_M^{(\lambda)} > r_M. \tag{39}$$

This shows that upweighting auxiliary-success trajectories increases expected main-task correctness under positive reasoning alignment.

A.7.3 Identical-Task Limit

We also consider the limiting case where the auxiliary task approaches the main task. Define

$$d(A, M) = \mathbb{E}_Z \left[|a(Z) - m(Z)| \right]. \tag{40}$$

Then

$$\begin{aligned} |r_A - r_M| &= |\mathbb{E}_Z[a(Z)] - \mathbb{E}_Z[m(Z)]| \\ &\leq \mathbb{E}_Z \left[|a(Z) - m(Z)| \right] = d(A, M). \end{aligned} \tag{41}$$

Therefore, if $a(Z) = m(Z)$ almost surely, then $d(A, M) = 0$ and

$$r_A = r_M. \tag{42}$$

This formalizes the intuition that when the auxiliary task and the main task are identical at the level of conditional success probabilities, their marginal success rates coincide.

If, in addition, the auxiliary and main outcomes are evaluated on the same trajectory so that $S_A = S_M$, then

$$\Pr(S_M = 1 \mid S_A = 1) = \Pr(S_A = 1 \mid S_A = 1) = 1. \tag{43}$$

Thus, in the strongest identical-task setting, auxiliary success becomes an exact certificate of main-task success.

A.7.4 Independent Attempts on Identical Tasks

A weaker identical-task setting occurs when the auxiliary and main tasks share the same conditional success probability but are sampled as conditionally independent attempts. Suppose

$$a(Z) = m(Z) = s(Z). \tag{44}$$

Then

$$r_A = r_M = r, \qquad r = \mathbb{E}_Z[s(Z)]. \tag{45}$$

Moreover,

$$\Gamma = \mathrm{Cov}_Z(s(Z), s(Z)) = \mathrm{Var}_Z(s(Z)). \tag{46}$$

Applying Theorem 4.2, we obtain

$$\Pr(S_M = 1 \mid S_A = 1) = r + \frac{\mathrm{Var}_Z(s(Z))}{r}. \tag{47}$$

Hence,

$$\Pr(S_M = 1 \mid S_A = 1) - r = \frac{\mathrm{Var}_Z(s(Z))}{r}. \tag{48}$$

Therefore, even for two conditionally independent attempts from the same task family, observing auxiliary success improves the posterior success probability of the main attempt whenever latent reasoning quality varies across trajectories.

A.8 Majority Voting Accuracy of RWR

We further evaluate the majority-voting performance of RWR under different numbers of sampled responses. For majority-voting evaluation, we set the sampling temperature to 0.8. We report maj@k, the accuracy of the final answer selected by majority voting over k sampled responses. As shown in Table 7, the accuracy improves steadily as the voting budget increases, suggesting that RWR-trained models can benefit from sampling multiple reasoning trajectories at inference time. The gains saturate at larger voting budgets, with maj@32 reaching 85.54% in all three runs.

A.9 Prompt Template

RWR uses an auxiliary-augmented prompt that requires the model to solve a deterministic auxiliary task before the main reasoning problem. The auxiliary answer is required to appear inside `<aux>...</aux>`, while the final main-task answer is required to appear inside `\boxed{...}`. The auxiliary answer is used for reward computation during training, whereas the main-task answer is used only for evaluation.

Table 7: Majority-voting accuracy of RWR under different sampling budgets. Values are percentages.

Metric	Mean $\pm$ Std	Min–Max	Three Runs
maj@1	32.93 ± 1.84	31.33–34.94	31.33/32.53/34.94
maj@4	64.26 ± 0.70	63.86–65.06	63.86/63.86/65.06
maj@8	73.09 ± 1.84	71.08–74.70	73.49/74.70/71.08
maj@16	80.32 ± 1.39	79.52–81.93	81.93/79.52/79.52
maj@32	85.54 ± 0.00	85.54–85.54	85.54/85.54/85.54

Table 8: Ablation study on test-time prompting. The verified warm-up prompt keeps the same auxiliary-augmented format as training and additionally provides the programmatically computed auxiliary answer, without revealing the main-task answer.

Test prompt	MATH500	GSM8K	AMC	AIME24
Main-only prompt	76.40	91.28	50.60	10.00
Verified warm-up prompt	80.80	93.18	55.42	16.67

Training prompt. During training, the model is given both the auxiliary task and the main problem. The model must generate the auxiliary answer by itself before solving the main problem.

Training prompt template

SYSTEM:
You are a helpful mathematical reasoning assistant.

USER:
You need to complete the following two tasks.

Task 1: Auxiliary task
[Auxiliary task description.]

Output format for Task 1:
Put only the auxiliary task's final answer inside `<aux>...</aux>`.

Task 2: Math problem
[Main mathematical reasoning problem.]

Output format for Task 2:
Think step by step, and put only the math problem's final answer inside `\boxed{...}`.

Test-time prompt. At test time, we consider two prompting protocols. The first is the standard main-only prompt, where the model directly solves the main mathematical problem. The second is the verified warm-up prompt, which keeps the same auxiliary-augmented structure as the training prompt but additionally provides the programmatically computed correct answer to the auxiliary task before the main problem. This answer is obtained by an exact auxiliary verifier and is independent of the main-task ground-truth answer. Therefore, the prompt supplies a verified warm-up signal without revealing the main-task label. We further study whether the verified warm-up context is useful at inference time for Qwen3 8B. As shown in Table 8, the main-only prompt removes the auxiliary task and directly asks the model to solve the mathematical problem, whereas the verified warm-up prompt provides the programmatically computed auxiliary answer before the main problem. The verified warm-up prompt achieves better performance on the mathematical benchmarks, indicating that the auxiliary context is not merely a training-time reward source, but can also serve as an effective inference-time warm-up signal. Since the auxiliary answer is computed automatically and is independent of the main-task answer, this protocol preserves the main-answer-withheld evaluation setting.

Verified warm-up test prompt template

SYSTEM:
You are a helpful mathematical reasoning assistant.

USER:
You are given a verified auxiliary task and a main mathematical problem.

Auxiliary task:
[Auxiliary task description.]

Verified auxiliary answer:
`<aux>[Programmatically computed auxiliary answer]</aux>`

Main math problem:
[Main mathematical reasoning problem.]

Output format:
Think step by step, and put only the math problem's final answer inside `\boxed{...}`.

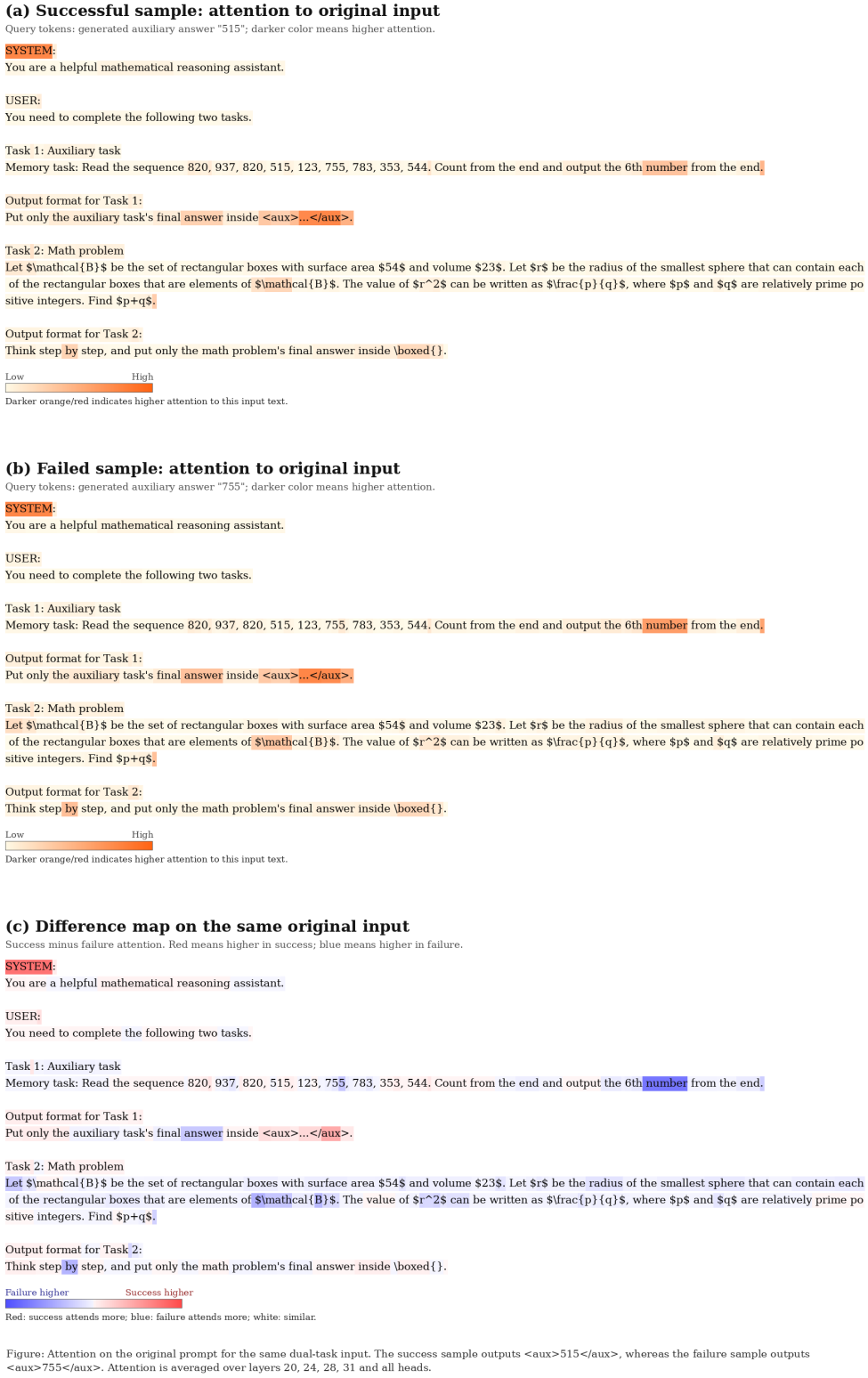

Figure: Attention on the original prompt for the same dual-task input. The success sample outputs <aux>515</aux>, whereas the failure sample outputs <aux>755</aux>. Attention is averaged over layers 20, 24, 28, 31 and all heads.

Figure 3: Attention over the original prompt for two sampled continuations under the same dual-task input. The successful continuation generates the correct auxiliary answer 515, whereas the failed continuation generates 755. The first two panels show the attention distribution for each continuation, and the third panel shows the signed difference between them.

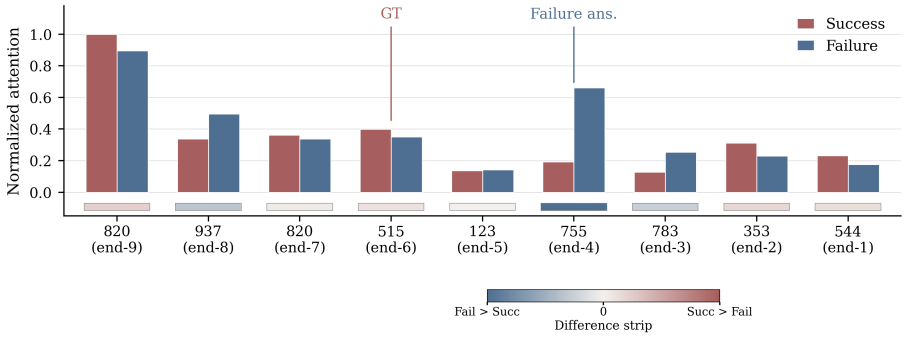

Figure 4: Zoomed attention over the numerical sequence in the auxiliary task. The failed continuation assigns substantially higher attention to `755`, the incorrect number it outputs, whereas the successful continuation outputs the correct answer `515`. The difference strip indicates whether attention is relatively higher in the successful or failed continuation for each input number.

A.10 Policy Learning and Inference-Time Warm-up

We evaluate Qwen3-8B-Base and the corresponding trained policies under two inference protocols: the main-task prompt alone and the same verified auxiliary warm-up prompt described in Appendix A.9. Table 9 reports both protocols for the base model, RWR, and the comparison methods.

Table 9: Qwen3-8B policy comparison under two inference prompts. Values are percentages; IFEval uses strict evaluation. Each panel uses the same inference prompt across policies.

Policy	MATH500	GSM8K	AMC	AIME24	IFEval	MMLU-Pro
(a) Main-only prompt						
Qwen3-8B-Base	72.40	27.82	20.93	3.75	50.89	52.92
RWR	76.40	91.28	50.60	10.00	58.78	52.79
Co-rewarding	80.80	92.42	53.46	14.48	60.70	57.95
Entropy	80.20	87.19	49.54	15.63	51.81	54.86
GT-Reward	82.60	87.26	54.22	17.15	52.78	57.11
Majority-Voting	79.80	89.76	49.09	15.83	51.80	56.93
Self-Certainty	80.20	80.74	50.75	15.73	50.98	54.17
(b) Verified auxiliary warm-up prompt						
Qwen3-8B-Base	73.60	28.81	20.48	6.67	47.13	48.92
RWR	80.80	93.18	55.42	16.67	55.82	53.14
Co-rewarding	77.40	93.18	50.60	13.33	50.28	49.67
Entropy	78.80	86.84	46.99	16.67	49.91	48.75
GT-Reward	83.00	86.89	56.63	10.00	49.72	54.07
Majority-Voting	81.00	89.63	48.19	16.67	48.61	53.42
Self-Certainty	79.00	80.22	53.99	20.00	46.77	51.25

Under main-only prompting, RWR improves over the base model on all four mathematical benchmarks, so its gains are not attributable solely to inference-time warm-up. Verified warm-up further improves these four results after RWR training, whereas applying it directly to the base model produces small and mixed changes. The relative performance of the trained policies varies across inference protocols and benchmarks.

A.11 Controls for Trajectory-Specific Verification

We compare RWR with two controls. *Shuffled correctness* permutes auxiliary verifier outcomes within each eight-rollout GRPO group, preserving the reward multiset while breaking its trajectory-specific correspondence. *Echo-token* retains the two-task structure and exact verification but replaces the auxiliary computation with a trivial token-copy task.

Table 10: Controls for the auxiliary verification signal. Values are percentages.

Policy	MATH500	GSM8K	AMC	AIME24	IFEval	MMLU-Pro
RWR	80.80	93.18	55.42	16.67	55.82	53.14
Echo-token	76.20	92.87	42.17	13.33	47.87	49.17
Shuffled correctness	22.00	16.38	4.82	0.00	52.68	49.50

Shuffling substantially reduces mathematical performance, indicating that preserving the reward distribution alone is insufficient. Echo-token also underperforms RWR on all six benchmarks. These controls support the importance of trajectory-specific feedback and reasoning-relevant auxiliary computation, beyond the two-task format or a trivial verifiable prefix alone.

A.12 Training Dynamics

We track main-task and auxiliary-task accuracy at seven checkpoints during a Qwen3-8B diagnostic run. Figure 5 shows the two accuracy trajectories. From step 0 to step 100, MATH main-task accuracy increases from 39.60% to 74.20%, while auxiliary accuracy increases from 28.60% to 91.40%. Main-task accuracy subsequently levels off at 76.20% as auxiliary accuracy remains high.

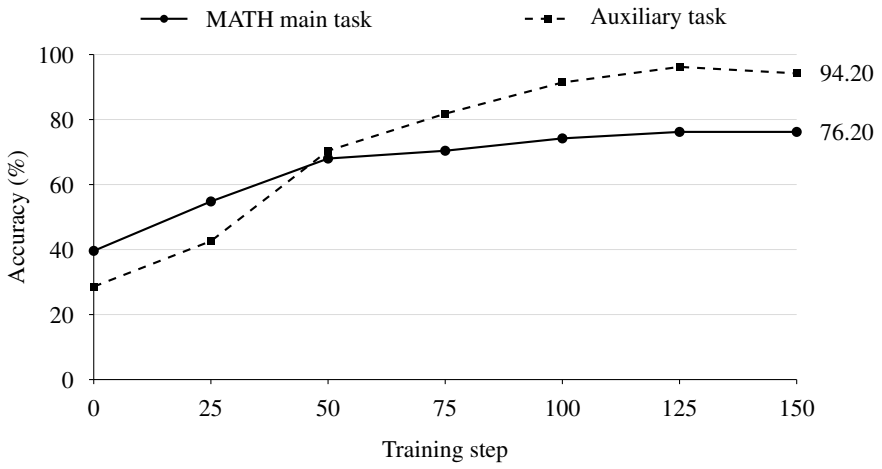

Figure 5: Checkpoint-level accuracies during a Qwen3-8B diagnostic run. Markers indicate the seven evaluated checkpoints; straight lines connect consecutive measurements.

A.13 Verified Warm-up and Auxiliary-Correct Selection

For the RWR-trained policy, we compare four inference protocols: the main task alone; providing the verified auxiliary answer before the main task; letting the model compute the auxiliary answer itself in a single rollout; and selecting self-computed trajectories whose auxiliary answers are correct.

Table 11: Verified auxiliary context versus self-computation and auxiliary-correct selection. Values are percentages.

Inference protocol	MATH500	GSM8K	AMC	AIME24
Main task only	76.40	91.28	50.60	10.00
Verified auxiliary answer + main	80.80	93.18	55.42	16.67
Self-computed auxiliary + main (pass@1)	74.40	92.49	50.60	16.67
Self-computed auxiliary + main (selection)	75.00	92.27	51.81	16.67

Providing the verified auxiliary answer improves all four benchmarks over main-only prompting without post-hoc trajectory selection. Relative to a single self-computed rollout, auxiliary-correct selection produces small, mixed changes. Thus, the benefit of verified auxiliary context cannot be explained by selection alone.

A.14 Auxiliary-Task Design Principles

We use three principles to guide auxiliary-task design. First, the task should exercise *shared reasoning primitives*: basic capabilities plausibly relevant to the target reasoning process, rather than directly solving the main problem. Second, it should allow *independent exact verification*: its answer should be independent of the main-task answer and deterministically checkable. Third, it should have *informative difficulty*, avoiding tasks that are either trivial or nearly impossible so that auxiliary correctness can retain useful reward variation during GRPO training.

The Memory, Sum, and Max comparison in Table 5 suggests that numerical similarity alone is insufficient. These principles remain heuristic, and effective auxiliary-task design may require domain expertise.

A.15 Preliminary Cross-Domain Evaluation

Code generation. We evaluate RWR on MBPP using independently generated, exactly verifiable Python state-tracking auxiliary tasks. These auxiliaries exercise a programming primitive shared with code generation and are independent of the MBPP solution and tests. Their difficulty is calibrated before training to avoid overly easy or overly difficult instances. MBPP accuracy increases from 58.4% before RWR training to 70.6% after training.

Meeting summarization. On MeetingBank-QA-Summary, we use the dataset-provided question–answer pairs as auxiliary tasks and meeting summarization as the main task. The training reward is derived from auxiliary-task correctness; the reference summary is not used as a training reward. RWR improves the three reported ROUGE scores, as shown in Table 12. Unlike the programmatically generated auxiliaries in the mathematical experiments, this setting uses existing QA supervision while withholding summary-level rewards.

Table 12: Preliminary results beyond mathematics. Left: MBPP accuracy (%). Right: MeetingBank-QA-Summary ROUGE scores. The two panels report separate task settings.

MBPP	Accuracy
Before RWR	58.4
After RWR	70.6

MeetingBank	ROUGE-1	ROUGE-2	ROUGE-L
Baseline	22.23	12.59	15.73
RWR	38.52	22.85	28.54

These results provide preliminary evidence for RWR in code generation and open-ended summarization.

A.16 Token and Wall-Clock Overhead

We measure incremental overhead using the same Qwen3-8B model, 100 training problems, eight rollouts per problem, a sampling temperature of 1.0, and one H20 GPU. Table 13 compares the main-only prompt with the auxiliary-augmented prompt.

Table 13: Token and wall-clock overhead in the controlled prompt comparison. Token counts are per rollout; time is per problem.

Prompt	Input tokens	Output tokens	Total tokens	Time (s)
Main only	88.1	445.9	534.0	1.874
Auxiliary + main	181.4	516.9	698.3	2.264

The auxiliary task increases processed tokens by 30.8% and measured wall-clock time by 20.8% in this comparison. The verifier remains deterministic and does not require a learned reward model. These measurements report the incremental overhead of auxiliary-augmented prompting under the stated rollout configuration.

A.17 Discussion

Conditioning versus optimization. Theorem 4.2 characterizes a fixed-policy conditioning effect under positive auxiliary–main alignment. RWR uses the auxiliary-derived reward for the entire auxiliary–main trajectory, including the main-task continuation. The fixed-policy analysis motivates the reward design, while the verification controls and training dynamics examine its empirical behavior during optimization. Proxy mismatch remains possible when auxiliary success ceases to reflect useful target-task reasoning.

Comparison with GT-Reward. Direct main-task rewards are a strong reference, but are not necessarily an empirical upper bound under finite-budget group-relative optimization. If all sampled responses receive the same reward, their reward-derived group-relative advantages are zero. Auxiliary correctness may still vary when all sampled main-task answers are incorrect, potentially retaining an informative optimization signal. Reward availability under a finite rollout budget offers a possible explanation for the observed gains over GT-Reward on some benchmarks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper provides a full set of assumptions and a complete proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data and code will be released once accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides details of implementation in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the statistical significance of the experiments in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper has discussed societal impacts in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code package and dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We communicate the details of the code as part of our submission via structured templates.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.