SARL: Optimizing the Path of Reasoning with Structure Aware Reinforcement Learning

Yifan Wang[1] Bolian Li[1] David Cho[1]
Ruqi Zhang[1] Fanping Sui[2] Ananth Grama[1]

[1]Purdue University [2]Texas Instruments

Abstract

Reinforcement learning is critical to improving large reasoning models, but its success relies heavily on verifiable rewards (RLVR), making it hard to use in open-ended domains where correctness is ambiguous and cannot be verified. Moreover, reasoning trajectories remain largely unconstrained, and optimizing solely towards the final answer can favor early exploitation over generalization. In this work, we investigate whether general reasoning ability can be improved by teaching models ***how to think*** (the structure of reasoning) rather than ***what to produce*** (the outcome of reasoning), and we extend traditional RLVR to open-ended settings. We introduce **S**tructure **A**ware **R**einforcement **L**earning (**SARL**), a label-free framework that constructs per-response reasoning maps from intermediate thinking steps and rewards their reasoning topology. SARL shifts supervision from destination to path, encouraging reasoning trajectories that are both locally coherent and globally efficient. On verifiable math tasks, SARL outperforms prior label-free RL baselines and even exceeds RL methods with ground truth supervision, with average gains of +9.1% under PPO and +11.6% under GRPO across four math benchmarks, with particularly large improvements on AIME25 (+35.5% with PPO and +44.7% with GRPO). On non-verifiable open-ended tasks, SARL achieves average gains of +34.6% under PPO and +30.4% under GRPO on WildBench across five task categories, outperforming prior label-free RL methods and DPO, which relies on additional preference labels. Beyond strong performance, SARL exhibits substantially lower KL divergence and higher policy entropy, indicating more stable and exploratory training dynamics. Code and data are available at `github.com/cacayaya/SARL`.

1 Introduction

Recently, Large Reasoning Models (LRMs) have demonstrated transformative capabilities in tackling complex, reasoning-intensive tasks, especially in mathematics and code generation [Guo et al., 2025, Yang et al., 2025]. Central to this progress is the Chain-of-Thought (CoT) paradigm, where models generate explicit, step-by-step reasoning before producing a final answer [Wei et al., 2022, Wang et al., 2022]. These reasoning capabilities are typically grounded in supervised fine-tuning (SFT) and, more crucially, in reinforcement learning (RL) [Ouyang et al., 2022, Shao et al., 2024]. Through RL training, models develop stronger reasoning abilities and exhibit sophisticated "thinking" behaviors such as self-reflection and multi-step planning [Lightman et al., 2024, Wang et al., 2024].

However, the current success of RL training is largely tethered to the scalability of Reinforcement Learning from Verifiable Rewards (RLVR) framework [Guo et al., 2025, Shao et al., 2024], which relies heavily on tasks with objective ground truths and automated verifiers to provide reward signals. This dependency creates a bottleneck of verifiability: in domains where ground truth is ambiguous,

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

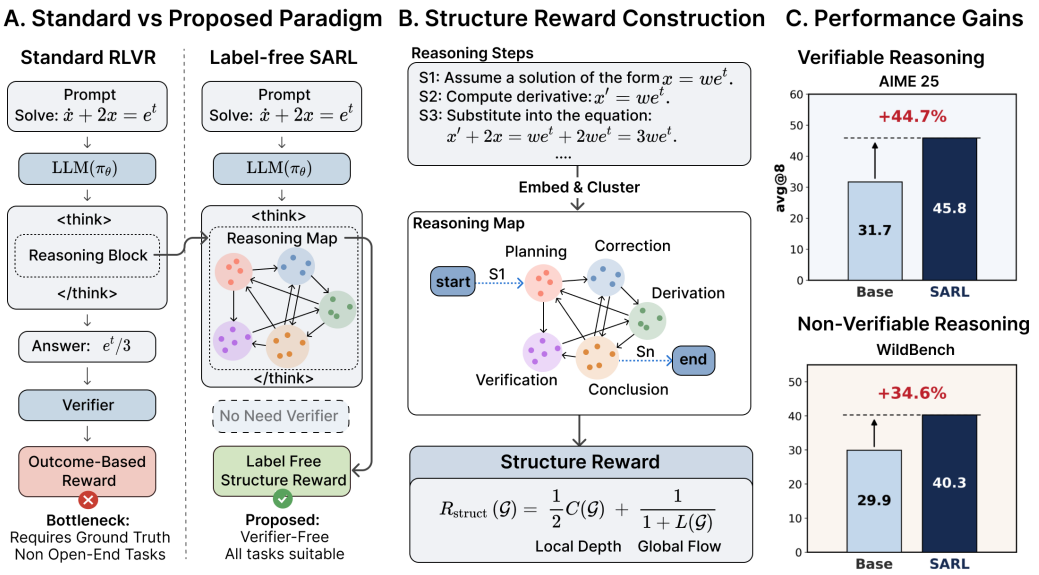


Figure 1: Overview of Structure Aware Reinforcement Learning (SARL). Left: SARL replaces outcome based supervision with a label free structure reward computed from the model's reasoning process. Middle: construction of the SARL Reasoning Map and Structure Reward. Right: Performance improvements on verifiable and open-ended tasks.

expensive to label, or non-existent, such as open-ended strategic planning or complex philosophical inquiry, reliable reward signals are largely unavailable.

Furthermore, even finer-grained approaches such as Process Reward Models (PRMs) [Lightman et al., 2024, Ma et al., 2023, Khalifa et al., 2025] remain fundamentally outcome-oriented: they assess how much each intermediate step contributes to reaching the correct final answer, rather than directly optimizing the structural organization of the reasoning process itself. As a result, the overall topology of reasoning trajectories remains unshaped. Moreover, PRMs are also prohibitively expensive to deploy in practice, as they require dense step-level annotations that are costly to collect and do not generalize to open-ended tasks without clear verification criteria. This raises a central question:

> *Can complex reasoning be optimized not through supervision derived from ground-truth answers, but by directly shaping how the model thinks?*

A compelling answer to this question is suggested by recent empirical work on the reasoning structural properties of Large Reasoning Models. Reasoning graphs extracted from LRMs exhibit measurable *small-world* topology [Watts and Strogatz, 1998], with dense local clustering coexisting with short global path lengths, and these structural features correlate positively with reasoning accuracy [Minegishi et al., 2025, Feng et al., 2025, Tan et al., 2025]. From the perspective of complex network theory, such organization balances local specialization with global integration, allowing efficient communication across distinct functional substructures of a reasoning trace. Together, these findings suggest that small-world topology is not merely a byproduct of good reasoning but a structural property that actively underlies it.

In this work, we introduce Structure Aware Reinforcement Learning (SARL), a label-free framework that takes this structural insight as a direct training signal. Rather than relying on ground-truth answers or step-level annotations, SARL constructs per-response reasoning maps from intermediate thinking steps and optimizes its small-world topology as a reward, enabling RL training across both verifiable and open-ended domains.

Our main contributions are as follows:

- We formulate the structural quality of reasoning trajectories as an intrinsic, label-free training signal, establishing a new optimization objective orthogonal to outcome-based rewards.

- We show that SARL is effective in both mathematical and open-ended reasoning, outperforming prior label-free baselines and even surpassing RL with ground-truth supervision.
- We observe that SARL yields lower KL divergence and higher policy entropy throughout training, suggesting more stable training dynamics and sustained exploration.

2 Related Work

Topology of Reasoning in Large Reasoning Models. A growing line of work characterizes reasoning in Large Reasoning Models (LRMs) through the structural organization of their reasoning trajectories. The *small-world* property [Watts and Strogatz, 1998], where dense local clustering coexists with short global path lengths, has emerged as a particularly informative way to characterize these trajectories. Minegishi et al. [2025] extract reasoning graphs from the hidden-state trajectories of LRMs and find that distilled reasoning models exhibit substantially higher cyclicity, diameter, and small-world index than base models, with these structural advantages scaling with task difficulty and correlating positively with accuracy. From a geometric perspective, Zhou et al. [2025] model chain-of-thought reasoning as smooth flows in the model's representation space, showing that logical transitions induce structured geometric trajectories governed by the semantics of intermediate propositions. At the level of trace analysis, Feng et al. [2025] introduce graph-based structural metrics for CoT traces and demonstrate that the fraction of abandoned reasoning branches, rather than trace length or self-review frequency, is the strongest predictor of correctness, providing causal evidence that structural organization shapes reasoning outcomes. Tan et al. [2025] further show that topological data analysis features of reasoning traces carry substantially higher predictive power for reasoning quality than standard graph-connectivity metrics. Collectively, these studies establish a robust empirical link between the structural organization of reasoning trajectories and reasoning quality. Our work takes the natural next step: rather than analyzing these structural properties post hoc, we optimize the small-world topology of reasoning graphs, built from individual steps as a direct intrinsic reasoning quality signal.

Label Free RLVR Reinforcement Learning from Verifiable Rewards (RLVR) has emerged as the dominant post-training paradigm for LRMs [Guo et al., 2025], using verifiable outcome rewards in domains where correctness can be automatically evaluated, primarily in mathematics and code. However, current RLVR-based training has two key limitations: (1) it relies on verifiable ground-truth rewards, making it inapplicable to open-ended domains such as creative writing, strategic planning, or advisory reasoning, where no automated verifier exists; and (2) it offers limited control over the structure of intermediate reasoning, as supervision operates solely at the level of correctness of final answer.Alternative label-free approaches have emerged to enable RLVR without ground-truth annotations. TTRL [Zuo et al., 2025] uses majority voting across rollouts to construct pseudo-labels as reward signals, which works for closed-form tasks but cannot handle open-ended settings where no consensus answer exists. Xin et al. [2025] propose surrogate rewards based on output format compliance and response length, though this remains restricted to math tasks with prescribed format templates. Meanwhile, EMPO [Zhang et al., 2025] and concurrent work [Agarwal et al., 2025] minimize the predictive entropy of model outputs, using the model's own uncertainty as a label-free reward signal. Similarly, Intuitor [Zhao et al., 2025] utilizes self-certainty scores measured via KL divergence to a uniform distribution as reward, and formulates the Reinforcement Learning from Internal Feedback (RLIF) framework. Beyond these methods, our approach derives the label free reward signal from the structure of the model's reasoning processes, and is applicable to not only verifiable tasks but also to open-ended and complex tasks in broader and real-world reasoning settings. Table 1 summarizes the key differences between these label-free approaches.

3 Methodology

In this section, we introduce Structure Aware Reinforcement Learning (SARL), a label free framework that rewards the topology of intermediate reasoning. We begin with the problem formulation, transitioning from outcome based RL to a structural training objective (Section 3.1). We then describe the construction of the reasoning map from thinking steps and the transitions of latent reasoning functions (Section 3.2). Finally, we define the structure reward (Section 3.3). The complete pipeline is summarized in Algorithm 1.

Table 1: Comparison of *label-free* RL methods. "Algorithm-agnostic" indicates whether the reward signal can be used with any RL algorithm (e.g., PPO) beyond group-based methods (e.g., GRPO).

Method	Reward Signal	Algorithm-agnostic	Open-ended	Verifiable
EMPO [Zhang et al., 2025]	Entropy minimization	✗	✓	✓
TTRL [Zuo et al., 2025]	Majority voting	✗	✗	✓
Format & Length [Xin et al., 2025]	Format + length	✓	✗	✓
SARL (Ours)	**Reasoning Structure**	✓	✓	✓

Algorithm 1 Structure Aware Reinforcement Learning

Require: Policy π_θ, embedding model $\mathcal{M}$, dataset $\mathcal{D}$, RL algorithm $\mathcal{A}$, rollouts per question N

for each training iteration **do**
  Sample batch $\{x_i\}$ from $\mathcal{D}$
  for each question x_i **do**
    Generate N trajectories $\{\tau_i^{(n)}\}_{n=1}^{N} \sim \pi_\theta(\cdot \mid x_i)$
    for each trajectory $\tau_i^{(n)}$ **do**
      Extract steps $\{s_t\}$ from <think> block; embed: $\mathbf{e}_t \leftarrow \mathcal{M}(s_t)/\|\mathcal{M}(s_t)\|$
      Cluster $\{\mathbf{e}_t\}$ into K nodes $\mathcal{V} = \{v_1, \ldots, v_K\}$; assign each s_t to $v(s_t) \in \mathcal{V}$
      Build $\mathcal{G}(\tau_i^{(n)})$: add edge $(v(s_t), v(s_{t+1}))$ for each $v(s_t) \neq v(s_{t+1})$
      Compute structure reward: $r_i^{(n)} \leftarrow \mathrm{SR}\Big(\mathcal{G}(\tau_i^{(n)})\Big)$
    end for
  end for
  Update $\theta \leftarrow \mathcal{A}\Big(\theta, \{(\tau_i^{(n)}, r_i^{(n)})\}\Big)$
end for

3.1 Problem Formulation

Let x denote an input question and let π_θ be a language model policy parameterized by θ. Following the chain-of-thought paradigm, the model produces a trajectory

$$\tau = (s_1, s_2, \ldots, s_T, a),$$

where $s_1, \ldots, s_T$ are intermediate reasoning steps inside the <think> block and a is the final answer. Standard RLVR optimizes an outcome level objective of the form

$$\max_\theta \ \mathbb{E}_{x\sim\mathcal{D},\ \tau\sim\pi_\theta(\cdot|x)}\big[\, r(a, x)\,\big], \tag{1}$$

where $r(a, x)$ is a verifiable reward that evaluates the final answer a against the question x, e.g., via exact match, execution, or format checking. While effective in domains where such verification is available, Eq. (1) conditions the reward entirely on the final answer and provides no direct signal about the quality of the reasoning process itself.

SARL replaces the outcome reward with a structural reward computed from the generated reasoning trajectory. For each rollout, we build a reasoning map $\mathcal{G}(\tau) = (\mathcal{V}, \mathcal{E})$ whose nodes represent latent reasoning functions and whose edges represent transitions between them. The SARL objective is

$$\max_\theta \ \mathbb{E}_{x\sim\mathcal{D},\ \tau\sim\pi_\theta(\cdot|x)}\big[\, \mathrm{SR}\big(\mathcal{G}(\tau)\big)\,\big], \tag{2}$$

where SR denotes the structure reward and measures the small-world organization of the resulting graph. Crucially, Eq. (2) does not rely on any verifier $r(a, x)$, allowing the policy to be trained even when correctness labels are unavailable or expensive to obtain.

3.2 Reasoning Map Construction

Step Extraction and Embedding. We first extract the content of the <think> block and segment it into individual reasoning steps, where each step corresponds to a coherent sentence naturally

delimited by the model's output formatting (e.g., \n\n for Qwen3-4B). Then, each step s_t is mapped to a unit normalized embedding $\mathbf{e}_t \in \mathbb{R}^d$,

$$\mathbf{e}_t = \frac{\mathcal{M}(s_t)}{\|\mathcal{M}(s_t)\|_2},$$

where $\mathcal{M}$ is a separate text embedding model.[1]

Latent reasoning functions. A reasoning trajectory typically consists of multiple reasoning functions, such as planning, derivation, calculation, verification, correction, and others. Importantly, the same reasoning function may appear repeatedly at different positions within a single trajectory. For example, multiple calculation steps may occur throughout the reasoning process, each operating on different intermediate quantities while serving the same functional role. SARL clusters the step embeddings $\{\mathbf{e}_t\}_{t=1}^T$ into K latent reasoning functions,

$$\mathcal{V} = \{v_1, v_2, \ldots, v_K\}. \tag{3}$$

Each node v_k represents one latent reasoning function shared by steps that perform similar operations, and we use $v(s_t) \in \mathcal{V}$ to denote the node assigned to step s_t. Clustering is performed independently for each response. We consider both KMeans and HDBSCAN for this step; the specific settings and the regimes in which each choice is more suitable are discussed in Appendix A.1.

Reasoning transitions. The edge set $\mathcal{E}$ captures transitions between reasoning functions. We add an undirected edge $(v(s_t), v(s_{t+1})) \in \mathcal{E}$ whenever two consecutive steps map to different nodes, i.e., $v(s_t) \neq v(s_{t+1})$. Repeated transitions between the same pair of reasoning functions contribute only one edge. We use an undirected graph because the reward is intended to measure structural organization and connectivity rather than directional control flow.

3.3 Structure Reward

Small-world prior. SARL is motivated by the observation that effective reasoning should balance local specialization with global integration. This suggests two desirable properties: high local clustering, which reflects coherent functional substructures, and short global path lengths, which reflect efficient transitions across different reasoning functions.

Let $\mathcal{N}(v_k)$ denote the neighbor set of node v_k, and let $\mathcal{V}_{\geq 2} = \{v_k \in \mathcal{V} : |\mathcal{N}(v_k)| \geq 2\}$ be the set of nontrivial nodes. We define the average clustering coefficient as

$$C(\mathcal{G}) \;=\; \frac{1}{|\mathcal{V}_{\geq 2}|} \sum_{v_k \in \mathcal{V}_{\geq 2}} \frac{\big|\{(v_i, v_j) \in \mathcal{E} : v_i, v_j \in \mathcal{N}(v_k)\}\big|}{|\mathcal{N}(v_k)|\,\big(|\mathcal{N}(v_k)| - 1\big)/2}, \tag{4}$$

which measures how strongly neighboring reasoning functions form locally coherent modules. We also define the average shortest path length over all reachable node pairs $\mathcal{P} = \{(v_i, v_j) : v_i \neq v_j,\ v_j \text{ reachable from } v_i\}$ as

$$L(\mathcal{G}) \;=\; \frac{1}{|\mathcal{P}|} \sum_{(v_i, v_j) \in \mathcal{P}} \delta(v_i, v_j), \tag{5}$$

where $\delta(v_i, v_j)$ denotes hop count distance. Lower values of $L(\mathcal{G})$ indicate more efficient communication across distinct reasoning functions.

The classical small-world index $\sigma = (C(\mathcal{G})/C_{\text{rand}})/(L(\mathcal{G})/L_{\text{rand}})$ [Watts and Strogatz, 1998], even under the analytical approximation $C_{\text{rand}} \approx \bar{k}/n$ and $L_{\text{rand}} \approx \ln n / \ln \bar{k}$ [Humphries and Gurney, 2008], becomes numerically unstable when $\bar{k} \approx 1$, precisely the sparse regime of the small reasoning graphs that arise in practice. We therefore define the structure reward (SR) as the equally weighted combination of the two classical small-world quantities, the clustering coefficient $C(\mathcal{G})$ and the average shortest path length $L(\mathcal{G})$, each mapped to $[0, \frac{1}{2}]$ so that they enter the reward on a common scale and the total reward is bounded in $[0, 1]$:

$$\mathrm{SR}(\mathcal{G}) \;=\; \frac{1}{2}\, C(\mathcal{G}) \;+\; \frac{1}{1 + L(\mathcal{G})}. \tag{6}$$

As shown in Eq. (4) and Eq. (5), $C(\mathcal{G})$ captures local specialization, while $L(\mathcal{G})$ captures global efficiency. Together, the two terms discourage traces that collapse into a single reasoning function or drift through long unstructured chains of transitions.

[1] We use `Qwen/Qwen3-Embedding-0.6B` for all experiments.

4 Experiments

In this section, we present empirical results to demonstrate the effectiveness of SARL. We first describe the setup (Section 4.1), then present results on verifiable mathematical reasoning (Section 4.2) and open-ended reasoning (Section 4.3), and finally provide analysis and ablation studies (Section 4.4).

4.1 Setup

Model and Datasets. We conduct all experiments using Qwen3-4B [Yang et al., 2025], a strong reasoning model that generates explicit reasoning trajectories within `<think>` blocks. We evaluate SARL under two distinct training regimes that cover both *verifiable* and *non-verifiable* reasoning domains. Dataset filtering and preparation details are provided in Appendix B.1.

- **Verifiable reasoning (Math).** We train on mathematical reasoning problems drawn from historical AIME competitions (1983–2024). Unlike simpler math tasks such as GSM8K [Cobbe et al., 2021] or MATH500 [Hendrycks et al., 2021], AIME problems demand long, complex reasoning, producing trajectories whose topological structure is rich enough to be meaningful, making them particularly well-suited for SARL training.
- **Non-verifiable reasoning (Open-ended).** We train on OpenRubrics-v2 [Liu et al., 2025], a large-scale preference dataset spanning diverse domains including creative writing, planning, coding, advice, and analytical reasoning. Unlike mathematical tasks, these problems lack explicit correctness labels.

Baselines. For the verifiable setting (Math Reasoning), we compare against two prominent label-free RL baselines: EMPO [Zhang et al., 2025] and TTRL [Zuo et al., 2025]. For the non-verifiable setting (Open-Ended Reasoning), we additionally compare against DPO [Rafailov et al., 2023] training on preference data. TTRL is not applicable to open-ended reasoning, as it relies on majority voting that assumes a consensus which may not exist in such settings.

Training and Evaluation. All reinforcement learning experiments are implemented using the veRL framework [Sheng et al., 2024]. Preference-based training with DPO is implemented using LlamaFactory [Zheng et al., 2024]. Full training details, including hyperparameters and optimization settings, are provided in Appendix B.2.

For evaluation, we assess mathematical reasoning performance on four benchmarks: MATH500 [Hendrycks et al., 2021], AIME25 [Balunović et al., 2025], AMC23 [knoveleng, 2025] and Minerva Math [Lewkowycz et al., 2022]. We adopt avg@8 as our primary evaluation metric, which provides a more stable assessment of model capability compared to single-run pass@1 scores. For open-ended reasoning evaluation, we employ the WildBench Elo rating system and task-category macro scores from the WildBench leaderboard [Lin et al., 2024].

4.2 Verifiable Reasoning (Mathematical Tasks) Results

We first validate the effectiveness of SARL in the well-established verifiable setting of mathematical reasoning, where RLVR methods are most commonly applied.

Structure reward closes the gap to ground-truth supervision. Table 2 shows that SARL matches or surpasses ground-truth RL under both PPO and GRPO, without ever observing a correctness label. Under GRPO, SARL achieves the best performance among all label-free RL methods, reaching an average relative improvement of +11.9%, outperforming EMPO (+10.8%) and TTRL (+10.2%), and even exceeding ground-truth RL (+11.1%). Under PPO, SARL achieves a relative improvement of +9.1%, again surpassing ground-truth PPO training (+8.8%). These results indicate that structural signals alone can provide sufficiently strong learning guidance to rival, and in some cases exceed, correctness-based supervision. Notably, unlike EMPO and TTRL, which rely on group-level optimization and are restricted to GRPO-style training, SARL generalizes across both PPO and GRPO frameworks. This algorithm-agnostic compatibility highlights its practical applicability across diverse reinforcement learning pipelines.

The largest gains from SARL training occur on **AIME25**, the most challenging benchmark requiring long multi-step mathematical derivations. Under GRPO, performance improves from 31.67 to

Method	AIME25	AMC23	MATH500	Minerva	Avg (Δ%)
Base	31.67	82.81	90.10	53.45	64.51
PPO					
w/ Ground-Truth†	41.67	86.56	92.75	59.74	70.18 (+8.8%)
w/ SARL (**Ours**)	42.92	85.00	92.53	61.08	70.38 (+9.1%)
GRPO					
w/ Ground-Truth†	**46.67**	84.38	93.15	**62.45**	71.66 (+11.1%)
w/ EMPO	44.58	86.56	93.23	61.44	71.45 (+10.8%)
w/ TTRL	42.91	86.56	93.15	61.86	71.12 (+10.2%)
w/ SARL (**Ours**)	45.83	**87.50**	**93.30**	61.99	**72.16 (+11.9%)**

Table 2: Results on mathematical reasoning benchmarks. **Bold** marks the best score in each column and underline marks the second best. † indicates methods that train with ground-truth correctness labels (oracle); all other methods are fully label-free. SARL achieves the best overall average under both algorithms and matches or surpasses the ground-truth supervision.

45.83, corresponding to a gain of +14.16 points (+45% relative improvement). Such improvements are substantially larger than those observed on easier benchmarks, like MATH500 and AMC23, suggesting that structural constraints become increasingly beneficial as reasoning becomes more complex.

4.3 Non-Verifiable Reasoning (Open-Ended Tasks) Results

A core motivation of our work is to enable reinforcement learning beyond verifiable domains using label-free structural rewards. To test this, we apply SARL on OpenRubrics-v2 [Liu et al., 2025], a diverse open-ended QA dataset.

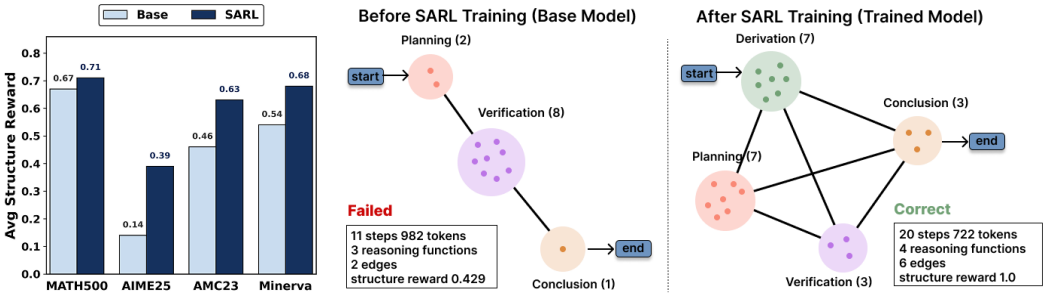
Method	Creative	Planning	Math	Info	Code	WB Score (Δ%)
Base	51.01	36.23	16.35	48.71	14.72	29.91
DPO						
w/ preference labels†	51.16	37.82	17.06	48.37	14.34	30.34 (+1.4%)
PPO						
w/ SARL (**Ours**)	**57.05**	**45.87**	**27.70**	**53.47**	**30.05**	**40.26 (+34.6%)**
GRPO						
w/ EMPO	51.01	36.11	17.62	45.69	12.92	29.20 (-2.4%)
w/ SARL (**Ours**)	55.34	43.56	26.83	51.98	29.91	39.01 (+30.4%)

Table 3: WildBench evaluation results for open-ended reasoning. Methods are grouped by training algorithm (DPO, PPO, GRPO). **Bold** marks the best score in each column and underline marks the second best. † indicates methods that train with explicit human supervision (preference labels for DPO); SARL and EMPO are fully label-free. TTRL is omitted because it requires binary correctness labels and does not generalize to open-ended tasks.

Structure reward enables reinforcement learning beyond verifiable domains. SARL achieves substantial performance gains on open-ended tasks without any labels or preference signals: under PPO, the WB Score improves from 29.91 to 40.26 (+34.6% relative improvement), and under GRPO to 39.01 (+30.4%). Note that TTRL requires binary correctness labels and format-based rewards [Xin et al., 2025] are math-specific, so neither generalizes here; we therefore compare against EMPO [Zhang et al., 2025] as the label-free baseline and DPO [Rafailov et al., 2023] as a reference with preference supervision. Even with this extra supervision, DPO yields only a marginal improvement (+1.4%), while EMPO slightly decreases performance (-2.4%). In contrast, SARL consistently delivers large gains across all five task categories.

As shown in Table 3, Planning improves by +26.6% (PPO) and +20.2% (GRPO), Creative writing by +11.8% and +8.5%, and Info/Advice by +9.8% and +6.7%. These categories have no single correct answer and rely on flexible judgment and knowledge synthesis, yet the structural prior still steers the model toward more coherent reasoning trajectories. The consistency of these gains suggests that small-world topology captures a fundamental quality of *how to think*, rather than serving as a proxy for correctness, and thus generalizes well beyond the verifiable domain.

4.4 Analysis and Ablations



(a) Structure Reward After Training. (b) Case study on a Minerva Math problem.

Figure 2: (a) Structure reward change across all benchmarks after SARL training. (b) Case study on a Minerva Math problem: reasoning topology before and after SARL training.

How SARL reshapes the reasoning map. Figure 2 analyzes how SARL reshapes reasoning structure on math benchmarks. Structure reward increases consistently across all four benchmarks after training, with the largest gain on AIME25 (+179%), where the base model severely under-reasons (∼21 steps) and SARL encourages substantially deeper deliberation. On MATH500, SR rises despite nearly unchanged step count, indicating that gains come from better *organization* of existing steps rather than more steps. These structural improvements co-occur with the accuracy gains in Table 2 while response length *decreases* (Table 4), confirming that SARL steers the model toward more efficient while effective reasoning. Figure 2(b) illustrates this concretely using a problem from Minerva Math: the base model produces a near-linear chain dominated by Verification (8 of 11 steps, SR=0.43, incorrect), while the trained model distributes reasoning across four functions with 6 edges (SR=1.0, correct) in *fewer* tokens (722 vs. 982).

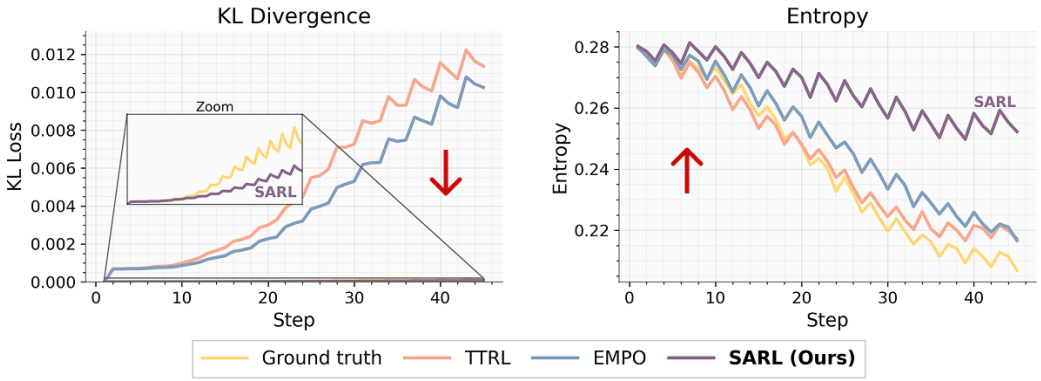


Figure 3: Training dynamics of different methods under GRPO.

Training dynamics analysis. Figure 3 compares the GRPO training dynamics of SARL with both label-free baselines (TTRL, EMPO) and ground-truth supervision on the verifiable math task. SARL consistently maintains the smallest KL divergence to the reference policy throughout training, while the other methods exhibit a substantially larger policy drift. This behavior reflects a common failure mode of label-free RL methods: excessive policy drift can destabilize optimization and eventually lead to model collapse.

At the same time, SARL maintains high policy entropy over most of training, indicating sustained exploration rather than early collapse to narrow reasoning patterns. In contrast, training directly on ground-truth rewards drives entropy drop quickly, suggesting premature exploitation that may hurt generalization. The central insight is that SARL improves reasoning through a stable optimization regime that combines minimal policy shift with relatively high exploration.

Length analysis. One possible concern is that the gains from SARL may come simply from generating longer responses. Table 4 shows that this is not the case. SARL substantially reduces average response length in both open-ended and mathematical reasoning settings, while achieving much higher performance. These results suggest that the benefit comes from producing more efficient and better organized reasoning traces rather than simply extending length. Full length comparisons against other baselines are provided in Appendix C.1.

Method	Avg Tokens	Δ vs. Base
Verifiable reasoning (Math).		
Base	5736.6	0.0
PPO w/ SARL	5075.6	-661.0
GRPO w/ SARL	4587.8	-1148.8
Non-verifiable reasoning (Open-ended).		
Base	2677.91	0.00
PPO w/ SARL	2303.17	-374.74
GRPO w/ SARL	2285.87	-392.04

Table 4: Response length comparison.

Reward variant	Avg (Δ%)
Base	64.51
SARL w/ L only	70.96 (+10.0%)
SARL w/ C only	71.97 (+11.6%)
SARL w/ $C + L$	**72.16 (+11.9%)**

Table 5: Ablation of the two structure reward terms (C: clustering coefficient; L: average shortest path length), averaged over MATH500, AIME25, AMC23, and Minerva.

Ablation: clustering coefficient vs. average shortest path length. The structure reward in Eq. (6) combines two classical small-world quantities of $\mathcal{G}$: the clustering coefficient $C(\mathcal{G})$ and the average shortest path length $L(\mathcal{G})$. We retrain Qwen3-4B under GRPO on the same math setting using only one of C or L as the reward. As shown in Table 5, each component alone yields substantial gains over the base model, indicating that both capture strong, independently useful structural signals for reasoning, which aligns with prior observations on their role in reasoning [Minegishi et al., 2025]. Combining them further improves performance, achieving the best average. Full results are provided in Appendix C.2.

5 Limitations

SARL is most effective for tasks that involve sufficiently long and complex reasoning processes. For straightforward tasks with only a few steps, the benefit of enforcing structural organization may be limited. Although the small-world prior works on the benchmarks studied here, other domains may favor different structures. The reward is a proxy for reasoning quality and can in principle be exploited; the best-of-n and graph-statistic checks in Appendix D.1 test particular failure modes but cannot establish the faithfulness or causal validity of individual traces. Performance also depends on step segmentation, embeddings, and clustering. Appendix D.3 reports checks with another embedding model, clustering method, model scale, and model family. Open-ended results rely on automated judges, whose scores vary by evaluator; we report a second benchmark and multiple WildBench judges in Appendix D.4. These limitations motivate more direct tests of reasoning faithfulness and broader robustness studies.

6 Conclusion

In this work, we introduced structure aware reinforcement learning (SARL), a label free framework that improves reasoning by rewarding the organization of the reasoning process rather than the correctness of final answers. By constructing Reasoning Maps and optimizing their small world topology, SARL provides a scalable training signal that applies to both verifiable and non verifiable tasks. Empirically, SARL matches or surpasses ground truth based RL on mathematical reasoning benchmarks, outperforms existing label free baselines such as EMPO and TTRL, and delivers large gains on open ended WildBench evaluation without using correctness labels or preference supervision. Our analysis shows that SARL maintains minimal policy drift and consistently higher entropy throughout training, indicating a more stable optimization regime with sustained exploration. Taken

together, these results suggest that reasoning structure itself is a strong and general learning signal, opening a path toward reinforcement learning beyond the bottleneck of verifiability.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025. URL https://matharena.ai/.

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

Yunzhen Feng, Julia Kempe, Cheng Zhang, Parag Jain, and Anthony Hartshorn. What characterizes effective reasoning? revisiting length, review, and structure of cot. *arXiv preprint arXiv:2509.19284*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Mark D Humphries and Kevin Gurney. Network 'small-world-ness': a quantitative method for determining canonical network equivalence. *PloS one*, 3(4):e0002051, 2008.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

knoveleng. Amc-23. https://huggingface.co/datasets/knoveleng/AMC-23, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*, 2024.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. *arXiv preprint arXiv:2510.07743*, 2025.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-level reward model as the navigators for reasoning. *arXiv preprint arXiv:2310.10080*, 2023.

Gouki Minegishi, Hiroki Furuta, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Topology of reasoning: Understanding large reasoning models through reasoning graph properties. *arXiv preprint arXiv:2506.05744*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Xue Wen Tan, Nathaniel Tan, Galen Lee, and Stanley Kok. The shape of reasoning: Topological analysis of reasoning traces in large language models. *arXiv preprint arXiv:2510.20665*, 2025.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 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, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Rihui Xin, Han Liu, Zecheng Wang, Yupeng Zhang, Dianbo Sui, Xiaolin Hu, and Bingning Wang. Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *arXiv preprint arXiv:2505.19439*, 2025.

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

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

Yufa Zhou, Yixiao Wang, Xunjian Yin, Shuyan Zhou, and Anru R Zhang. The geometry of reasoning: Flowing logics in representation space. *arXiv preprint arXiv:2510.09782*, 2025.

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

A Implementation Details

A.1 Clustering details for functional nodes

We consider two practical choices for clustering step embeddings into latent reasoning functions: KMeans and HDBSCAN.

KMeans. KMeans is well suited to settings where the number of reasoning steps is not very large and the step structure is relatively regular. In our experiments, we use KMeans for the mathematical reasoning setting, where trajectories often follow a more regular progression such as setup, derivation, verification, and answer extraction, and we find KMeans empirically stable and reproducible across runs. Since KMeans requires specifying the number of clusters, we use $k \approx \sqrt{M}$, where M is the number of reasoning steps in the current response.

HDBSCAN. HDBSCAN is preferable when the step structure is more heterogeneous because it can infer the number of clusters automatically and can better capture non spherical cluster geometry. In our experiments, we use HDBSCAN for the open ended setting, where the reasoning patterns are more diverse across prompts. For HDBSCAN, we use Euclidean distance, i.e., $d(\mathbf{e}, \mathbf{e}') = \|\mathbf{e} - \mathbf{e}'\|_2$. We set min_cluster_size $= \max(2, \min(5, M/4))$ and min_samples $=$ min_cluster_size $- 1$, where M is the total number of step embeddings. Noise points (assigned label -1) are either grouped into a separate cluster or, if all points are noise, assigned to a default cluster.

B Experimental Setup

B.1 Data Preparation

AIME Historical Problems (Math Setting). We use competition problems from the American Invitational Mathematics Examination (AIME) spanning 1983 to 2024, sourced from the publicly available *AIME Problems 1983–2024* dataset on Kaggle.[2] This dataset contains 933 problems in total. Each problem is formatted as a single-turn prompt with the instruction "Let's think step by step and output the final answer within \boxed{}." appended to guide the model to solve the problem and box the final answer.

OpenRubrics-v2 (Open-Ended Setting). We use the `OpenRubrics/OpenRubric-v2` dataset from HuggingFace,[3] which contains preference pairs across diverse task categories including creative writing, planning, coding, and advice.

We use only the `instruction` field as the training prompt and apply the following filtering pipeline to select high-quality prompts. First, we perform response length filtering: since this dataset contains reference responses, we use response length as a proxy to identify questions that elicit sufficiently complex reasoning. We require both reference responses should be longer than 512 tokens. Responses shorter than 512 tokens likely correspond to trivial questions that do not require deep reasoning. This filtering ensures we select prompts that generate appropriately complex reasoning traces. Second, we apply deduplication by removing duplicate instructions to prevent duplicate training samples. After filtering, we retain 3294 samples.

B.2 Training Configuration

All training are mainly implemented via the `verl` library [Sheng et al., 2024], though we also have TRL [von Werra et al., 2020] version. All experiments are conducted on a single node equipped with 8 NVIDIA H100 GPUs using Fully Sharded Data Parallel (FSDP2) for model parallelism.

[2] https://www.kaggle.com/datasets/tourist800/aime-problems-1983-to-2024
[3] https://huggingface.co/datasets/OpenRubrics/OpenRubric-v2

The structure reward requires step embeddings computed from each rollout's `<think>` block. To avoid interfering with the main policy rollout (which runs under vLLM), we launch a standalone embedding server using `vllm.entrypoints.openai.api_server` with the `Qwen/Qwen3-Embedding-0.6B` model on a dedicated port, consuming only 5% of one GPU's memory. The reward function queries this server via HTTP for each generated trajectory.

Tables 6 and 7 summarize the training hyperparameters used in our PPO and GRPO experiments, respectively. Across both algorithms, the Math and Open-Ended settings mainly differ in the training data, maximum response length, total epochs, and clustering method.

Hyperparameter	**Math Setting**	**Open-Ended Setting**
RL algorithm	PPO	PPO
Advantage estimator	GAE	GAE
Training data	AIME 1983–2024	OpenRubrics-v2
Base model	`Qwen/Qwen3-4B`	`Qwen/Qwen3-4B`
Rollout engine	vLLM	vLLM
Rollouts per prompt (N)	8	8
Train batch size	256	256
Policy mini batch size	256	256
Policy micro batch size / GPU	32	32
Critic micro batch size / GPU	32	32
Actor learning rate	1×10^{-6}	1×10^{-6}
Critic learning rate	1×10^{-5}	1×10^{-5}
Max prompt length	1,024 tokens	1,024 tokens
Max response length	8,192 tokens	4,096 tokens
Filter overlong prompts	True	True
Truncation mode	`error`	`error`
Data shuffle	False	False
KL loss coefficient	0.001	0.001
KL loss type	`low_var_kl`	`low_var_kl`
Entropy coefficient	0	0
Use KL in reward	False	False
Critic warmup	0	0
Total epochs	15	5
Save frequency	20	20
Test frequency	5	5
Parallelism strategy	FSDP2	FSDP2
Dynamic batch sizing	True	True
Gradient checkpointing	True	True
Activation offload	True	True
Rollout GPU memory utilization	0.7	0.7
Embedding model	`Qwen/Qwen3-Embedding-0.6B`	`Qwen/Qwen3-Embedding-0.6B`
Embedding server	standalone vLLM API server	standalone vLLM API server
Clustering method	KMeans	HDBSCAN

Table 6: PPO training hyperparameters for the Math and Open-Ended settings.

C Additional Ablations

C.1 Full Response Length Comparison

For completeness, we report the full response length comparisons against all available baselines for both the math and open ended settings.

Hyperparameter	Math Setting	Open-Ended Setting
RL algorithm	GRPO	GRPO
Training data	AIME 1983–2024	OpenRubrics-v2
Base model	`Qwen/Qwen3-4B`	`Qwen/Qwen3-4B`
Rollout engine	vLLM	vLLM
Rollouts per prompt (N)	8	8
Train batch size	256	256
Policy mini batch size	256	256
Policy micro batch size / GPU	32	32
Actor learning rate	1×10^{-6}	1×10^{-6}
Max prompt length	1,024 tokens	1,024 tokens
Max response length	8,192 tokens	4,096 tokens
Filter overlong prompts	True	True
Truncation mode	`error`	`error`
Data shuffle	False	False
KL loss coefficient	0.001	0.001
KL loss type	`low_var_kl`	`low_var_kl`
Entropy coefficient	0	0
Use KL in reward	False	False
Critic warmup	0	0
Total epochs	15	5
Save frequency	20	20
Test frequency	5	5
Parallelism strategy	FSDP2	FSDP2
Dynamic batch sizing	True	True
Gradient checkpointing	True	True
Activation offload	True	True
Rollout GPU memory utilization	0.7	0.7
Embedding model	`Qwen/Qwen3-Embedding-0.6B`	`Qwen/Qwen3-Embedding-0.6B`
Embedding server	standalone vLLM API server	standalone vLLM API server
Clustering method	KMeans	HDBSCAN

Table 7: GRPO training hyperparameters for the Math and Open-Ended settings.

Method	AIME25	AMC23	MATH500	Minerva	Avg	Δ vs. Base
Base	7728.0	5884.1	4182.6	5151.7	5736.6	0.0
PPO w/ Ground-Truth	7047.7	4634.7	3124.9	4021.7	4707.3	-1029.3
PPO w/ SARL	7296.6	5069.4	3451.1	4485.4	5075.6	-661.0
GRPO w/ Ground-Truth	6477.7	3956.6	2559.0	3143.8	4034.3	-1702.3
GRPO w/ EMPO	6651.5	4058.6	2702.4	3327.9	4185.1	-1551.5
GRPO w/ TTRL	6635.0	4057.8	2663.5	3258.5	4153.7	-1582.9
GRPO w/ SARL	6984.9	4506.9	3029.4	3829.9	4587.8	-1148.8

Table 8: Full average response length comparison on mathematical reasoning benchmarks.

C.2 Structure Reward Ablation Details

We retrain Qwen3-4B under GRPO on the AIME math training set using each component of the structure reward in Eq. (6) alone, while keeping all other settings identical. Both components individually yield substantial improvements over the base model. This observation is consistent with prior work that identifies clustering and path length as key structural properties associated with strong reasoning behavior [Minegishi et al., 2025].

Method	Avg Tokens	Δ vs. Base
Base	2677.91	0.00
DPO	2661.46	-16.45
EMPO	2731.92	+54.01
PPO w/ SARL	2303.17	-374.74
GRPO w/ SARL	2285.87	-392.04

Table 9: Full average response length comparison on WildBench.

Reward variant	AIME25	AMC23	MATH500	Minerva	Avg (Δ%)
Base	31.67	82.81	90.10	53.45	64.51
SARL w/ L only	40.42	**87.81**	92.92	**62.68**	70.96 (+10.0%)
SARL w/ C only	**45.83**	87.19	93.00	61.86	71.97 (+11.6%)
SARL w/ $C + L$	**45.83**	87.50	**93.30**	61.99	**72.16 (+11.9%)**

Table 10: Per-benchmark ablation of the two structure reward terms under GRPO on the math setting. C denotes the average clustering coefficient and L denotes the average shortest path length, with the reward components $C(\mathcal{G})/2$ and $1/(1 + L(\mathcal{G}))$ defined in Eq. (6).

D Reward Robustness Analysis

This section examines the structural reward, its combination with outcome supervision, and the robustness of the results across model and evaluation choices. Unless specified otherwise, math columns are accuracy percentages and "Avg" is the mean over AIME25, AMC23, MATH500, and Minerva. Relative changes in parentheses are measured against the corresponding base model. The training comparisons use single runs; we do not attach statistical significance claims to small differences.

D.1 Structural Reward Diagnostics

Best-of-n proxy stress test. For each of the 30 AIME25 problems, we sampled 256 responses from the base Qwen3-4B policy and 256 from the SARL-trained policy (15,360 responses in total). For each candidate pool, we selected the response with the highest structure reward and checked its answer using an independent ground-truth verifier. We averaged over 1,000 random permutations per problem. Random selection and oracle pass@n provide reference points. Table 11 shows that selected structure reward rises with pool size while verifier accuracy increases substantially through $n = 256$, with a small fluctuation between $n = 64$ and $n = 128$ for the trained policy. There is no sustained accuracy collapse over this tested range. The oracle gap for the trained policy remains substantial, so this test does not show that the proxy identifies every correct response.

Policy	n	Random acc.	SR-selected acc.	Oracle pass@n	Selected SR
Base	1	22.97	22.97	22.97	0.165
	8	22.64	36.98	38.51	0.298
	32	22.53	44.47	45.74	0.379
	64	22.94	47.39	47.86	0.417
	128	22.56	49.19	49.22	0.449
	256	22.89	50.00	50.00	0.473
SARL	1	41.90	41.90	41.90	0.354
	8	41.99	55.56	61.00	0.575
	32	41.94	59.40	69.69	0.713
	64	41.99	59.97	72.32	0.767
	128	41.88	59.77	74.92	0.802
	256	41.97	60.00	80.00	0.817

Table 11: Best-of-n selection on AIME25. Accuracy and oracle pass@n are percentages; SR is structure reward.

Graph statistics. Table 12 compares all 7,680 responses per policy from the same AIME25 samples. Mean density increases after training but remains well below one, and complete graphs occur in 0.143% of trained-policy responses. These observations address collapse toward fully connected graphs in this setting, but do not by themselves establish semantic faithfulness.

Metric	Base	SARL
Mean structure reward	0.164	0.354
Mean graph density	0.129	0.278
Full-clique rate (%)	0.000	0.143
Normalized clique number	0.110	0.243
Transitivity	0.135	0.288
State-revisit rate	0.200	0.438

Table 12: Topology of sampled AIME25 responses before and after SARL training.

D.2 Combining Structural and Outcome Rewards

We also trained with an equally weighted sum of ground-truth accuracy reward and structure reward on math. Table 13 compares the combined objective with either reward alone. The combination has the highest PPO average (71.06), whereas SARL alone has the highest GRPO average (72.16). The differences are small relative to the gains over Base and lack multi-seed uncertainty estimates. This experiment applies to verifiable math only; open-ended prompts have no ground-truth accuracy reward.

Method	AIME25	AMC23	MATH500	Minerva	Avg (relative Δ)
Base	31.67	82.81	90.10	53.45	64.51
PPO w/ Ground-Truth	41.67	86.56	92.75	59.74	70.18 (+8.8%)
PPO w/ SARL	42.92	85.00	92.53	61.08	70.38 (+9.1%)
PPO w/ Ground-Truth + SARL	42.92	86.39	93.60	61.31	71.06 (+10.1%)
GRPO w/ Ground-Truth	46.67	84.38	93.15	62.45	71.66 (+11.1%)
GRPO w/ SARL	45.83	87.50	93.30	61.99	72.16 (+11.9%)
GRPO w/ Ground-Truth + SARL	43.75	87.50	93.20	61.86	71.58 (+11.0%)

Table 13: Math accuracy under separate and combined rewards. The combined reward is 0.5 ground-truth accuracy reward $+\,0.5$ structure reward.

D.3 Sensitivity to Representation and Model Choice

Table 14 reports GRPO math results with HDBSCAN in place of KMeans, and with `microsoft/harrier-oss-v1-0.6b` in place of `Qwen/Qwen3-Embedding-0.6B`, changing one component at a time. Both alternatives retain substantial gains over Base. In the reported training run, GRPO with SARL took 11 h 27 m 28 s versus 11 h 0 m 26 s for ground-truth reward training on the stated hardware (Appendix B.2).

Variant	Setting	AIME25	AMC23	MATH500	Minerva	Avg (relative Δ)
Base	–	31.67	82.81	90.10	53.45	64.51
Clustering	HDBSCAN	43.33	86.88	92.53	61.67	71.10 (+10.2%)
Clustering	KMeans	45.83	87.50	93.30	61.99	72.16 (+11.9%)
Embedding	Qwen3-Embedding-0.6B	45.83	87.50	93.30	61.99	72.16 (+11.9%)
Embedding	harrier-oss-v1-0.6b	43.75	87.50	92.71	61.86	71.46 (+10.8%)

Table 14: Clustering and embedding ablations for Qwen3-4B under GRPO on math. The KMeans and Qwen embedding rows are the same reference run.

We further evaluated Qwen3-8B and Llama-3.2-3B-Instruct. For the non-thinking Llama model, we used reasoning-unit segmentation of its generated text, preserving equations, lists, steps, and cases, rather than requiring an explicit `<think>` block. Tables 15 and 16 report the corresponding math and WildBench results. Gains transfer to these tested models, although the Llama math gain from SARL is smaller than its ground-truth reward gain. The WildBench score is a weighted macro score over five categories; relative changes use the base score for each model family.

Model	Method	AIME25	AMC23	MATH500	Minerva	Avg (relative Δ)
Qwen3-8B	Base	31.67	78.13	87.93	54.60	63.08
	GRPO w/ Ground-Truth	41.67	86.25	93.13	63.92	71.24 (+12.9%)
	GRPO w/ SARL	40.00	85.63	92.75	62.55	70.23 (+11.3%)
Llama-3.2-3B-Instruct	Base	0.00	17.50	39.43	17.88	18.70
	GRPO w/ Ground-Truth	0.42	23.44	45.10	24.40	23.34 (+24.8%)
	GRPO w/ SARL	1.67	22.50	39.75	18.66	20.65 (+10.4%)

Table 15: Math results across model scales and families.

Model	Method	Creative	Planning	Math	Info	Coding	WB Score (relative Δ)	Length
Qwen3-8B	Base	55.87	43.05	24.84	53.51	24.72	37.39	11170.99
	DPO	55.13	41.97	23.17	52.51	25.47	36.72 (-1.8%)	11084.02
	GRPO w/ SARL	60.73	50.70	33.81	59.01	36.70	45.83 (+22.6%)	9419.47
Llama-3.2-3B	Base	39.69	28.01	10.71	33.17	11.00	21.93	3802.48
	DPO	-5.65	-14.25	-32.46	0.62	-30.46	-19.10 (-187.1%)	3008.39
	GRPO w/ SARL	40.26	29.61	15.40	34.60	13.08	24.18 (+10.3%)	2276.47

Table 16: WildBench results across model scales and families. Negative WB Scores are possible because the score is centered on the judge's mid-level rating. Length is average tokens.

D.4 Open-Ended Evaluation Robustness

We evaluated Qwen3-4B on AlpacaEval as a second open-ended benchmark (Table 17). We also re-scored WildBench outputs using three judges (Table 18). We report the standard errors accompanying the AlpacaEval metrics. WildBench scores are not directly comparable across judges, so the second table gives ranks and absolute changes from Base within each judge. Both SARL variants rank first or second under each judge, but the size of their gains depends on the evaluator. These automated evaluations do not replace human assessment.

Model	Win rate (%)	SE (win)	LC win rate (%)	SE (LC)	Avg length
Base	38.39	1.72	24.04	0.08	6733
DPO	38.81	1.72	24.27	0.10	6577
PPO w/ SARL	46.26	1.76	30.37	0.12	5332
GRPO w/ EMPO	39.75	1.73	27.56	0.12	7186
GRPO w/ SARL	43.41	1.75	26.97	0.17	5187

Table 17: Qwen3-4B on AlpacaEval. LC denotes length-controlled win rate; SE denotes reported standard error.

	Rank			WB Score Δ vs. Base		
Method	GPT-4o	GPT-5.6-sol	Claude-5-sonnet	GPT-4o	GPT-5.6-sol	Claude-5-sonnet
Base	4	3	3	0.00	0.00	0.00
DPO	3	5	5	+0.43	-0.21	-7.98
GRPO w/ EMPO	5	4	4	-0.71	-0.05	-7.07
GRPO w/ SARL	2	1	2	+9.10	+5.34	+1.82
PPO w/ SARL	1	2	1	+10.35	+4.93	+4.36

Table 18: Qwen3-4B WildBench sensitivity to the judge. Rank 1 is best. DPO uses preference labels; Δ is an absolute, within-judge difference from Base.

E Ethics and Broader Impact Statement

The paper does not involve human-subject data collection, personally identifiable information, or deployment in safety-critical settings. Potential risks include enabling stronger reasoning models that may also be misused in broader downstream applications. However, the paper primarily contributes to a label-free RL training technique rather than a new capability domain. We document datasets, implementation details, hyperparameters, and compute requirements, supporting external scrutiny while discouraging inappropriate use.