StaRPO: Stability-Augmented Reinforcement Policy Optimization

Jinghan Zhang[1], **Fengran Mo**[2], **Tharindu Cyril Weerasooriya**[3], **Ruimin Dai**[1], **Xiaoyan Han**[1], **Yanjie Fu**[4], **Dakuo Wang**[5], **Kunpeng Liu**[1]*

[1]Clemson University, [2]Rochester Institute of Technology, [3]Accenture (United States), [4]Arizona State University, [5]Northeastern University

jinghaz@clemson.edu, frmvcs@rit.edu, kunpenl@clemson.edu

Abstract

Reinforcement learning (RL) is effective in enhancing the accuracy of large language models in complex reasoning tasks. Existing RL policy optimization frameworks rely on final-answer correctness as feedback signals and rarely capture the internal logical structure of the reasoning process. Consequently, the models would generate fluent and semantically relevant responses but logically inconsistent, structurally erratic, or redundant. To this end, we propose **StaRPO**, a stability-augmented reinforcement learning framework that explicitly incorporates reasoning stability into the optimization objective. Our StaRPO decomposes stability into two computable lightweight metrics: the *Autocorrelation Function* (ACF) to evaluate local step-to-step coherence, and *Path Efficiency* (PE) to evaluate global goal-directedness of the reasoning trajectory. These stability rewards are combined with task rewards to provide complementary and process-aware feedback. We validate the effectiveness of using ACF and PE rewards by showing their correlation with logic errors on two backbone models. Experiments on four reasoning benchmarks show that StaRPO consistently outperforms compared baselines and can enhance both final-answer accuracy and logical stability.

1 Introduction

Recent advances in large language models (LLMs) for reasoning-intensive tasks demonstrate remarkable effectiveness. In particular, reinforcement learning (RL) has significantly raised the performance ceiling of LLMs on complex problem-solving benchmarks Guo et al. [2025], Team [2024], Liu et al. [2024], Pennino et al. [2025], Bandyopadhyay et al. [2025] with appropriate optimization approaches.

However, reasoning is not merely about producing a correct final answer. From the perspective of human users, ideal reasoning should consist of three components: an accurate final result in terms of completing the task; a logically sound reasoning process to support the explainability and reliability of the result; and proper knowledge grounding for factual correction to improve the reliability and truthworthiness of the model's decisions and outputs. Although final-answer accuracy and factual correctness can usually be clearly quantified and used as feedback signals in previous studies Zhang et al. [2024], the logical soundness is a higher-level property describing the correctness of inferential relations between premises and conclusions Govier et al. [2010]. This is because logic does not have a quantitative formulation, but it is useful and widely recognized by human evaluators. In complex real-world applications, such as multi-round decision-making, mathematical theorem proving, and medical consultations, LLMs often produce reasoning chains that, while fluent and contextually relevant, remain logically inconsistent or structurally erratic. Such flawed reasoning must be mitigated, as it risks providing users with spurious justifications or misleading conclusions.

*Corresponding author.

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

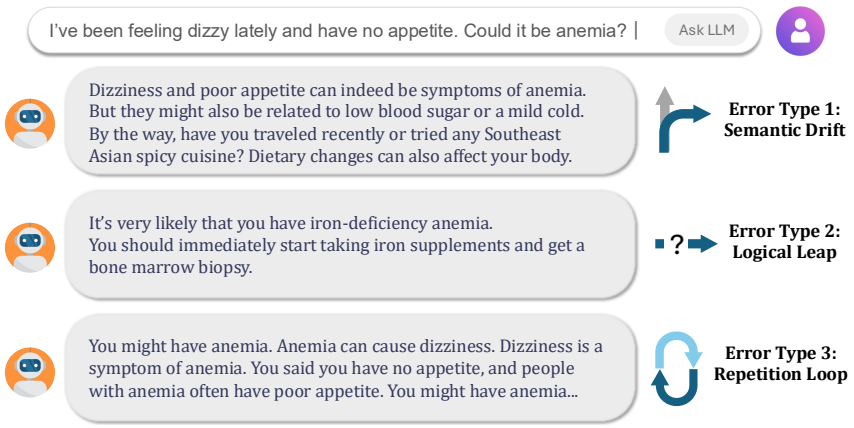


Figure 1: Example of three types of logical errors.

In Figure 1, we provide three representative types of logical errors in a medical diagnosis scenario. The first response digresses from the core question to travel history. The second one skips the necessary step of inferring the disease from the given symptoms and directly goes to providing medical advice, while the third response falls into a redundant loop. All of these answers are semantically relevant to the query and expressed with a confident tone, yet none of them exhibit a coherent logical structure where conclusions are derived from preceding premises George [1995]. In the representation space of potential responses, this phenomenon reflects that logically valid answers actually constitute only a small subset of the broader representation space of semantically plausible responses Yang et al. [2024]. From a distributional perspective, generating logically valid responses is considerably more challenging than producing outputs that are merely semantically coherent Anderson et al. [2024]. These logical errors are difficult to detect in language models with confidence metrics (e.g., entropy or log-probabilities), even with available ground-truth answers, as they arise from deeper issues in how the model navigates and structures its internal reasoning trajectory Sun et al. [2023].

Although most result-based RL frameworks can significantly improve LLM reasoning performance Shao et al. [2024], Feng et al. [2025], Lin et al. [2025], Zheng et al. [2025], their logical stability in terms of reasoning task are not satisfactory by solely relying on final-answer correctness as feedback signals. The follow-up studies Zhang et al. [2025e], Chen et al. [2025a] incorporate multiple types of rewards or signals for the reasoning process training, but still overlook the specific reasoning steps and abnormal logic within.

In this study, we aim to augment the logical correctness of the LLM reasoning process, i.e., the **logical stability** of reasoning. Logical stability contains three aspects: (1) **semantic coherence**: local step-to-step consistency in meaning, (2) **logical coherence**: the absence of discontinuities, leaps, or contradictions, and (3) **goal alignment**: maintaining a consistent trajectory toward the target answer. To this end, we build on the RL framework, with Group Relative Policy Optimization (GRPO) to incorporate additional trajectory-level signals and allow comparison of diverse reasoning behaviors.

Our target. We target calculating and rewarding reasoning stability metrics effectively during training. Specifically, the stability is monitored from two perspectives: (1) **local coherence**, which ensures that each reasoning step logically follows from the previous one; (2) **global progression**, which ensures that the overall reasoning trajectory steadily advances toward a coherent final answer. Since reasoning stability is hard to quantify and is not a parametric property, we adopt RL as an alignment framework rather than using explicit supervision. Moreover, to avoid the high cost of using "LLM-as-a-judge" Gu et al. [2024] for constructing feedback signals, we monitor the abnormal performance of reasoning trajectories in the embedding level as lightweight metrics for efficient computation.

Our method. To this end, we introduce **StaRPO**, a **Sta**bility-Augmented **R**einforcement **P**olicy **O**ptimization framework. We quantify reasoning stability by decomposing it into two measurable metrics and converting them into rewards. These stability rewards are combined with task performance rewards to provide feedback for reasoning learning as supplementary. Specifically, we introduce the *Autocorrelation Function (ACF)* and *Path Efficiency (PE)* as two complementary metrics, where

ACF evaluates the local stability by measuring directional changes at the embedding level, and PE evaluates the global stability by calculating the overall efficiency of the reasoning trajectory.

We conduct experiments to first evaluate the effectiveness of ACF and PE as indicators of logical stability. By examining the correlation between abnormal values of these metrics and three types of reasoning errors, we demonstrate that their combination provides a reliable signal of instability. We then conduct experiments with StaRPO on several reasoning benchmarks. Our experiments based on Qwen2.5-1.5B/7B-Instruct Team [2024] models show that StaRPO consistently outperforms GRPO, where the generated reasoning steps are more accurate and reliable.

2 Related Work

2.1 RL for Reasoning in LLMs

Recent studies have demonstrated that RL can significantly enhance the reasoning capabilities of LLMs [Liu et al., 2025a, Zhang et al., 2025b] by taking the autoregressive generation as a sequential decision-making process and enabling the LLMs to optimize long-horizon behaviors through delayed reward signals [Sun et al., 2023, Zhang et al., 2025a]. Recent research has successfully applied RL across various domains, including mathematical reasoning, task decomposition, and agentic systems. These studies report significant and consistent gains in both output precision and overall task success rates [Yue et al., 2025, Lin et al., 2025]. Given that RL provides a robust framework for optimizing reasoning trajectories, we leverage it to enhance the logical consistency and inferential stability of LLM-generated outputs.

2.2 Process-Level Rewards for Reasoning

Moving beyond conventional outcome-based optimization, several studies have integrated auxiliary signals to refine the generation process. These process-oriented rewards encompass metrics such as response length, structural compliance, and information entropy. Furthermore, recent research has incorporated external verifier models or critics to provide a posterior evaluation of reasoning quality [Wu et al., 2025b, Yu et al., 2025, Chen et al., 2025b, Zhang et al., 2025f]. Process Reward Models (PRMs) provide explicit feedback by evaluating intermediate reasoning steps [Lightman et al., 2024, Wang et al., 2024, Li and Li, 2025, Cheng et al., 2025]. These process-level rewarding approaches enrich the signal space and correct some factorial error in sentence-level reasoning steps [Zhang et al., 2025d, Cho et al., 2025]. However, many process-level reward methods focus on step-level correctness or a step's contribution to the final answer, e.g., Rewarding Progress [Setlur et al., 2025] scores a step by how much it increases a separate prover policy's probability of completing the solution correctly. These rewards discourage incorrect steps but do not necessarily constrain the coherence between steps or the structure of the full reasoning trajectory. Some methods also use separate reward models or verifiers to score generated steps or complete trajectories, which would increase the computational cost. Moreover, reward models may require task-specific training, which can limit their transfer to new tasks. We therefore aim to develop a method that complements correctness rewards by encouraging local coherence and global progress, without relying on a separately trained, task-specific reward model.

2.3 Representation-Space Signals for Reasoning

To quantify the logic differences among the potential generated texts in terms of a given sequence, existing studies use distances and directions of the embedding vectors as geometric measures of logical similarity and change [Luus et al., 2021, Nguyen et al.]. This is because in distributed representations, relative positions and directional variations often reflect transitions between logic states [Huang et al., 2022, Patil et al., 2023].

In the context of autoregressive generation, the representations of the consecutive reasoning step form an embedding trajectory. Existing studies show that there is a correlation between the properties of trajectory (like the smoothness and consistency) [Fang et al., 2016, Hosseiny Marani and Baumer, 2023] and the semantic coherence and structural stability of generated text. Conversely, abrupt or irregular directional changes within the trajectory are frequently associated with substantial semantic state changes [Wu et al., 2025a, Chen et al., 2025c]. In the meantime, the logical errors frequently stem from unstable state transitions and anomalous semantic representations [Tang et al., 2023, Liu

et al., 2025b]. To this end, we adopt the embedding of abnormal directional changes as signals to track potential logical instability during reasoning.

3 Preliminary and Motivation

In this paper, we take the autoregressive language model parameterized as a stochastic policy π_θ. Let x denote a query (prompt) and $\mathcal{D}$ be the query set, for a response $y = (y_1, \ldots, y_{|y|})$ to query x, the policy likelihood is:

$$\pi_\theta(y \mid x) = \prod_{t=1}^{|y|} \pi_\theta\left(y_t \mid x, y_{<t}\right), \tag{1}$$

where $y_{<t} = (y_1, \ldots, y_{t-1})$ and $|y|$ is the response length. The reward model assigns a reward score $r(x, y) \in [0, 1]$ to each query–response pair. In this paper, all expectations are taken over $x \sim \mathcal{D}$ and responses sampled from an "old" policy $\pi_{\theta_{\text{old}}}$ unless stated otherwise.

Group Relative Policy Optimization (GRPO). GRPO [Shao et al., 2024] estimates advantages through relative comparisons among multiple responses to the same query. Compared with PPO [Schulman et al., 2017], GRPO enhances the scalability and robustness of training and is value-free, as it does not need a value function [Guo et al., 2025]. GRPO computes the normalized advantages within a response group without KL regularization, whose optimization goal is:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(w_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\Big)\right] \tag{2}$$

where G is the group size. The importance ratio $w_{i,t}(\theta)$ and advantage $\hat{A}_{i,t}$ of all token $y_{i,t} \in y_i$ are:

$$w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}, \quad \hat{A}_{i,t} = \hat{A}_i = \frac{r(x, y_i) - \text{mean}\big(\{r(x, y_j)\}_{j=1}^{G}\big)}{\text{std}\big(\{r(x, y_j)\}_{j=1}^{G}\big)}. \tag{3}$$

GRPO enhances the training stability and solves the reward sparsity by normalizing rewards among the group. It allows the policy to learn from relative comparison when the absolute reward signals are weak or missing. However, since the advantage $\hat{A}_i$ is computed solely with the reward of response, the signals are meaningful only when there is a sufficiently informative reward difference among the group. If all the responses in a group are wrong or similarly suboptimal, the relative advantage would not have a positive impact. Moreover, GRPO's optimization target is still based on the final output. Logically unstable reasoning trajectories are not identified and can not be used as signals to further distinguish the responses in a group.

Motivation. In the context of using RL training to enhance models' reasoning capability, a crucial question is *how to take advantage of the reasoning trajectory to provide additional feedback signals?* With these signals, the policy can improve the quality of reasoning paths while maintaining high final-answer accuracy. Besides, when the responses in one group have similar or equally-poor outcomes, these complementary signals can differ the responses and thus mitigate the limitation of the group-relative reward mechanism.

Given the potential effectiveness of these signals for policy optimization, another key question is: *what kind of signal can effectively reflect the logical stability of a reasoning trajectory?* Meanwhile, considering the practical deployment in large-scale RL training, these signals should also be lightweight, easy to compute, and stable across different tasks and model scales, without relying on additional supervision or expensive external models. To explore these questions, we explore the signals that can indicate logic-level changes and design a corresponding mechanism to merge the signals into the RL optimization process.

4 StaRPO: Stability-Augmented Reinforcement Policy Optimization

In this section, we introduce StaRPO, a stability-augmented reinforcement policy optimization framework for improving reasoning reliability for LLMs. We first define the logical-stable space

and the reasoning trajectory's coherent evolution within logical constraints. Then, we introduce two complementary metrics to quantify local. With these metrics, we extend the GRPO framework by incorporating stability-augmented rewards into the optimization objective.

4.1 Logical-Stable Space

As shown in Figure 2, at any decoding step t, given a query x and the generated prefix $y_{<t}$, an autoregressive model can determine a conditional distribution $\pi_\theta(\cdot \mid x, y_{<t})$. For an ideal multi-step reasoning process, the model would sample one token per step from the conditional distribution. This sequence of token-level decisions is formulated as a series of state transitions in semantic space, which is used to form a reasoning trajectory from x to a correct answer region A.

The support of the token's conditional distribution constitutes the token generation space, denoted as $\mathcal{S}_t$. This space characterizes all token choices that the model assigns a non-negligible probability under the current generation context. In $\mathcal{S}_t$, there exists a subset $\mathcal{S}_t^{\text{sem}} \subset \mathcal{S}_t$, which we refer to as the **semantic-stable space** $\mathcal{S}_t^{\text{sem}}$. As the working procedure of most well-trained LLMs, the tokens sampled from this region can make coherent, fluent, and semantically plausible text. However, as shown in Figure 1, the output text from this semantic-stable space is acceptable at the language level, but does not necessarily guarantee logical stability. In the semantic-stable space $\mathcal{S}_t^{\text{sem}}$, there exists a much smaller and more restrictive subset $\mathcal{S}_t^{\text{log}} \subset \mathcal{S}_t^{\text{sem}}$, denoted as **logical-stable space**. In this space, the tokens not only maintain semantic coherence but also keep state transitions that are consistent with the logical structure of the task. If the generated sequence satisfies $y_t \in \mathcal{S}_t^{\text{log}}$ at every decoding step, the reasoning trajectory can progress continuously under logical constraints. At the textual level, each subsequent inference step builds upon its predecessors, progressing incrementally toward the final target answer A.

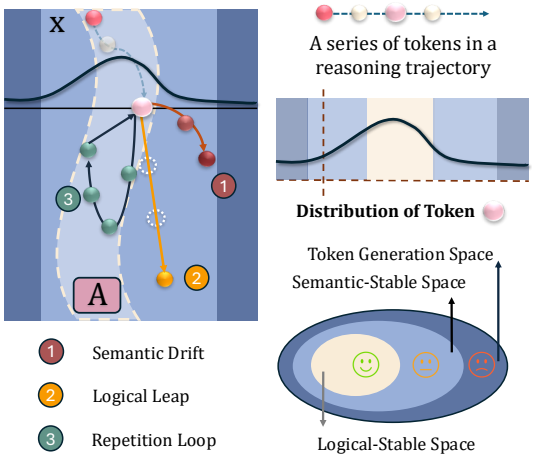


Figure 2: Logical-stable space in token generation space for a reasoning trajectory. From a query x, the LLM generates a sequence of token-level transitions targeting the correct answer region A step-by-step. Incorrect logic may result in three types of errors.

If at any step t the generated token satisfies $y_t \notin \mathcal{S}_t^{\text{log}}$, the reasoning trajectory may shift from the intended direction and produce a logically wrong generation. The three types of logical errors occur when the trajectory goes into the semantic-stable space but not in the logical-stable subspace: (1) The red trajectory corresponds to semantic drift, that is, individual statements may remain fluent, the overall trajectory go wrong direction and not to a path to the reasoning target; (2) The yellow trajectory corresponds to logical leap, that is, the model skips several logically necessary intermediate states and leaps to a semantically far away region; (3) The green trajectory corresponds to repetition loop, that is, the generation lost direction and wonders within a local semantic neighborhood.

4.2 Stability-Augmented Rewards

In the last section, we characterized the reasoning process as an evolving token-level trajectory and analyzed how it can fail in logical instability. We now turn to the question of how such failures can be alleviated or prevented during RL training.

In practice, it is hard to directly define and reward correct and stable logic even with an RL framework. Logical correctness is an abstract, high-level property that often depends on holistic human judgment and is hard to formalize or automatically evaluate at scale. In contrast, identifying unstable logical errors is much easier. Thus, our aim to introduce appropriate penalties in the reward signal when errors are highly likely exists and guides models back to a logical stable space.

In addition, reasoning stability exists at both local and global levels. Local level emphasizes continuity and consistency, which requires continuous reasoning states to evolve smoothly and consistently without sudden leaps or big direction changes. Global level requires that the overall trajectory should make meaningful progress toward solving the target problem, instead of stalling or looping redundantly in the semantic space. Thus, we design our StaRPO method from both local and global perspectives and the corresponding stability-augmented feedback signals to constrain and optimize the stability of model reasoning.

4.2.1 Local: Stepwise Autocorrelation Reward

To capture local level stability, we segment the LLM's reasoning generation into K steps $\{Step_k\}_{k=1}^{K}$ and represent each step $Step_k$ by a semantic embedding vector h_k which is the average of token-level hidden states within the step. We evaluate the evolution of reasoning as the embedding changes through steps. We use the difference between consecutive step embeddings as a proxy for the local transition direction:

$$\Delta h_k = h_k - h_{k-1}, \quad k = 2, \ldots, K \tag{4}$$

Δh_k is expected to capture how the semantic state moves among reasoning steps. A stable reasoning process should exhibit consistent directional changes across steps, whereas reversals and large directional deviations usually occur alongside unstable logical behavior, e.g., sudden leaps that skip necessary intermediate states. In contrast, persistently small turning angles over many steps can also be an abnormal sign, as it may reflect repetitive updates that do not introduce new information.

Intuitively, we adopt an autocorrelation-based metric to quantify the consistency between consecutive transition directions. Autocorrelation is originally a concept in time-series analysis to measure temporal dependence between adjacent observations in a sequence, and captures whether successive states evolve consistently over time. Here we draw an analogy by treating a reasoning trajectory as an ordered sequence of semantic state transitions. The consecutive transitions are analogous to adjacent observations in a time series. Specifically, we define the stepwise autocorrelation function (**ACF**) as the cosine similarity between adjacent changes as:

$$\text{ACF}_k = \cos(\Delta h_k, \Delta h_{k-1}) = \frac{\langle \Delta h_k, \Delta h_{k-1} \rangle}{\|\Delta h_k\| \cdot \|\Delta h_{k-1}\|}. \tag{5}$$

Moderately positive ACF_k values indicate locally consistent progression, and values that are too low (weak or negative alignment) or excessively high (near-identical updates repeated over steps) are often associated with unstable behaviors such as logical leaps or repetition loops. Then, the ACF reward of a reasoning chain is the average over all steps as Eq. 6, which encourages coherent progression between steps and penalizes abrupt semantic jumps or redundancy.

$$r_{\text{acf}} = \frac{1}{K-1} \sum_{k=2}^{K} \cos\big(\Delta h_k, \Delta h_{k-1}\big). \tag{6}$$

4.2.2 Global: Path Efficiency Reward

Simply using ACF to capture the local consistency between adjacent reasoning steps is not enough. A reasoning process could be locally consistent but fail globally, like wandering in semantic space or repeating without getting close to the solution. To address this limitation, we introduce a complementary metric that evaluates global stability from a goal-directed progression aspect.

A globally stable reasoning trajectory should transit efficiently from the initial state toward the final state without unnecessary detours. Based on this intuition, we quantify global progression by comparing the net displacement of the trajectory with its total path length. Specifically, we define the net displacement D to measure how far the reasoning state moves from the beginning to the end, and then the total path length L can be calculated accordingly. This path length L accumulates all intermediate semantic transitions along the trajectory. Then, the path efficiency η is formulated in Eq. 7, where ϵ is a small constant for numerical stability.

$$D = \|h_K - h_1\|, \quad L = \sum_{k=2}^{K} \|h_k - h_{k-1}\|, \quad \eta = \frac{D}{L + \epsilon}, \quad \eta \in [0, 1], \tag{7}$$

A value $\eta \approx 1$ indicates a straight, goal-directed trajectory (stable reasoning), while $\eta \to 0$ reflects oscillations or loops (unstable reasoning). We use the path efficiency (**PE**) as the global stability

reward in Eq. 8. This reward explicitly encourages reasoning trajectories to make steady, goal-aligned progress toward the final answer, while penalizing locally fluent but globally inefficient trajectories.

$$r_{\text{pe}} = \frac{\|h_K - h_1\|}{\sum_{k=2}^{K} \|h_k - h_{k-1}\| + \epsilon}. \tag{8}$$

4.3 StaRPO Objective

On top of the GRPO framework, we incorporate reasoning stability feedback into the policy optimization. For GRPO, the advantage of each response y_i is computed based on its relative outcome-level reward within a group of responses to the same query. In StaRPO, we extend this formulation by adding a stability-augmented reward for each reasoning trajectory as Eq. 9, where $\lambda_{\text{acf}}, \lambda_{\text{pe}} \geq 0$ control the trade-off between task success and stability.

$$r_{\text{StaRPO}}(x, y_i) = r(x, y_i) + \lambda_{\text{acf}} \cdot r_{\text{acf}} + \lambda_{\text{pe}} \cdot r_{\text{pe}}, \tag{9}$$

Accordingly, we compute group-relative advantages with the stability-augmented rewards. For a group of G responses $y_{i}{}_{i=1}^{G}$ generated for the same query x, the advantage for response y_i is:

$$A_i = \frac{r_{\text{StaRPO}}(x, y_i) - \text{mean}\big(\{r_{\text{StaRPO}}(x, y_j)\}_{j=1}^{G}\big)}{\text{std}\big(\{r_{\text{StaRPO}}(x, y_j)\}_{j=1}^{G}\big)}. \tag{10}$$

where all tokens within the same response share the same scalar advantage A_i. Ultimately, the overall objective function of StaRPO is formulated as:

$$J_{\text{StaRPO}}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D} \\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min\Big(w_{i,t}(\theta) A_i,\ \text{clip}\big(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big) A_i \Big) \right]. \tag{11}$$

5 Experiments

5.1 Experiment Setup

Evaluation Benchmarks and Metrics. We use four reasoning benchmarks to evaluate our StaRPO, including: (1) GSM8K [Cobbe et al., 2021], which contains grade-school math word problems testing multi-step arithmetic reasoning; (2) MATH [Hendrycks et al., 2021] contains competition-level math questions; (3) GPQA-Diamond [Rein et al., 2024], a curated subset of GPQA, which contains 198 PhD-level science questions authored by domain experts in physics, chemistry, and biology; and (4) AIME24 [Math-AI, 2024] contains 30 competition-style math problems covering arithmetic, algebra, and geometry from the American Invitational Mathematics Examination. We use final-answer accuracy **(Result)** as well as the process accuracy **(Process)**, which measures the proportion of reasoning trajectories with no detected logical errors, as evaluation metrics.

Baselines. We compare our method with two types of baselines. The first category constrains model behavior by modifying RL training objectives, including (1) **GRPO** [Shao et al., 2024], which optimizes policies via group-relative outcome comparison without a value model; (2) **CPPO** [Lin et al., 2025], which improves GRPO efficiency by pruning low-advantage completions during training; and (3) **λ-GRPO** [Wang et al., 2025], which mitigates GRPO's length bias by learning adaptive token-level weights without introducing additional process-level supervision. Another one category consists of methods that directly supervise or intervene in the reasoning process using process-level signals, including (4) **Entropy**-based control [Zhang et al., 2025c], which guides reasoning depth based on model uncertainty estimates; and (5) LLM-**Planner** [Song et al., 2023], which leverages external planning modules to explicitly guide intermediate decision making during task execution.

Implementation Details. All RL-based methods are implemented following the previous studies [Lin et al., 2025, Wang et al., 2025] with the same training configuration to ensure fair comparison. Unless otherwise specified, the group size is fixed to 8, and all models are trained for two RL iterations with a KL regularization coefficient $\beta = 0.04$. We conduct our experiment on `Qwen2.5-Instruct` in two sizes, 1.5B and 7B [Qwen Team, 2024]. The sampling temperature is set to 1.0, except for GSM8K as 0.9, following common practice [Lin et al., 2025], and we extract step representations

from the last Transformer layer, i.e., `hidden_states[-1]`, for each backbone model for this experiment. The learning rate is set to $1e-6$. For StaRPO, we estimate the empirical distributions of ACF and PE metrics using the first 100 generated reasoning trajectories for each benchmark, and use the corresponding tail regions as the initial abnormality ranges for both metrics. All experiments are conducted on four NVIDIA H200 80G GPUs.

Model	Method	GSM8K		MATH		GPQA		AIME		Avg.	
		Result	Process	Result	Process	Result	Process	Result	Process	Result	Process
Qwen 1.5B	Original	55.72	52.39	47.73	43.94	19.19	16.67	3.33	3.33	31.49	29.08
	CPPO	76.80	73.62	54.55	52.08	34.85	28.79	**13.33**	**10.00**	44.88	41.12
	λ–GRPO	78.17	76.95	53.98	52.65	33.33	30.81	6.67	6.67	43.04	41.77
	GRPO	73.92	72.02	53.60	49.62	26.77	23.23	10.00	**10.00**	41.07	38.72
	Entropy	65.05	64.29	52.65	50.57	25.25	23.23	6.67	6.67	37.41	36.20
	Planner	60.35	58.83	49.81	47.35	22.73	19.70	0.00	0.00	33.22	31.47
	StaRPO	**79.68**†	**78.54**†	**56.25**†	**56.06**†	**35.35**†	**34.85**†	10.00	**10.00**	**45.32**†	**44.86**†
Qwen 7B	Original	89.46	88.63	53.60	52.27	28.79	26.77	10.00	10.00	45.46	44.42
	CPPO	90.52	88.93	74.43	71.97	**38.89**	35.35	23.33	20.00	56.79	54.06
	λ–GRPO	90.90	89.08	75.38	74.62	37.88	34.85	16.67	13.33	55.21	52.97
	GRPO	90.30	89.39	74.24	73.11	35.86	33.33	23.33	16.67	55.10	53.13
	Entropy	90.37	89.23	67.61	65.72	30.30	29.80	13.33	13.33	50.40	49.52
	Planner	89.76	89.08	52.08	50.57	28.79	24.24	6.67	6.67	44.33	42.64
	StaRPO	**91.66**†	**91.43**†	**77.46**†	**76.89**†	38.38†	**37.88**†	**26.67**	**23.33**	**57.71**†	**57.38**†

Table 1: Performance comparison across four tasks. The best and second-best performance is set in dark and light blue. Superscript † denotes significant improvements with paired t-test at $p < 0.05$ over GRPO.

5.2 Overall Performance

We first conduct standard training and evaluation across four benchmarks for both final-answer accuracy and process accuracy. As shown in Table 1, we observe that StaRPO consistently achieves the best or second-best performance across all datasets and backbone models, especially in knowledge-intensive tasks (GPQA) and long-horizon reasoning tasks (MATH). Specifically, StaRPO gains more improvements in process accuracy. For example, on Qwen-1.5B, StaRPO improves process accuracy over GRPO by $+11.62\%$ on GPQA, which is 3.04% higher than the improvement in final-answer accuracy. This observation suggests that the stability-augmented rewards primarily enhance the logical quality of the reasoning process rather than solely increasing the chance of producing a correct answer. About the compared baselines, although GRPO and those that modify RL optimization mechanisms (CPPO and λ-GRPO) can improve final-answer accuracy by providing stable and efficient optimization signals, these methods cannot provide stable enhancement in process accuracy as our StaRPO. The average improvement for CPPO and λ-GRPO over GRPO in process accuracy is $+2.4\%$ and $+3.05\%$, respectively, while our StaRPO provides $+6.14\%$ on Qwen-1.5B. Besides, the process accuracy of CPPO and λ-GRPO even drops based on the 7B model on GSM8K. Entropy-based control and planning methods achieve smaller improvements, and we can see degradation on complex tasks. This phenomenon suggests that relying on uncertainty signals or an external planning module without optimization policies is difficult to provide stable and scalable improvements in reasoning quality. In contrast, StaRPO directly incorporates stability-augmented signals into the RL reward, enabling consistent and robust enhancements in both process and final-answer accuracy.

```
Question: Use numbers and basic arithmetic operations (+ - * /) to obtain 24.
Example 1 (Semantic Drift): Input: 5 5 5 9  Model Output: 5 * 5 = 25, 25 / 9 = 2.7778,
2.7778 / 9 = 0.309 ...
Example 2 (Logical Leap): Input: 5 5 5 9 Model Output: 19 + 5 = 24
Example 3 (Repetition Loop): Input: 8 4 3 1  Model Output: 8 / 4 = 2, 2 * 4 = 8, 8 / 4 =
2, 2 * 4 = 8 ...
```

Figure 3: Example of logical reasoning errors on *Game of 24*.

Error Type	ACF Low	ACF High	PE (t-test)	PE (M-W)
Semantic Drift	0.0217*	0.0207*	0.0358*	0.1363
Logical Leap	0.0063**	0.0011**	0.0251*	0.0483*
Repetition	0.2158	0.0297*	0.0408*	0.0379*

Table 2: Significance test between reasoning errors and stability metrics. Superscript * and ** denotes $p < 0.05$ and $p < 0.01$, respectively.

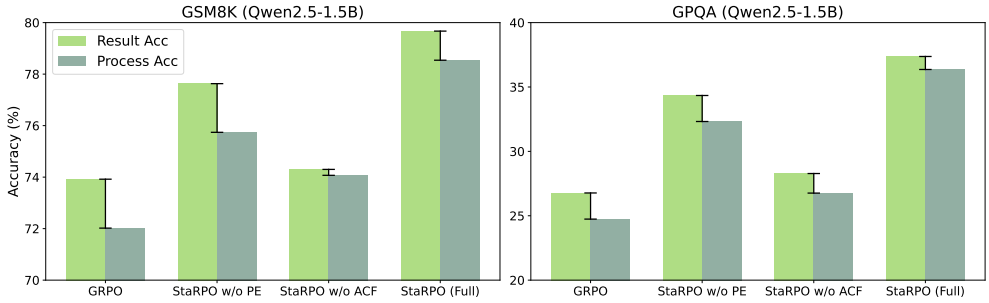


Figure 4: Ablation studies with result and process accuracy across different model variants on two datasets.

5.3 Correlation between Logical Errors and Stability Metrics

In this experiment, we investigate whether the abnormal values of the two proposed metrics, ACF and PE, are significantly correlated with the three types of logical errors (semantic drift, logical leap, and repetition loop) as introduced in Figure 1. We conduct this experiment on Task *Game of 24* [Yao et al., 2023], which requires the model to obtain the target value 24 by applying basic arithmetic operations to a given set of numbers (*"Use numbers and basic arithmetic operations (+ − × ÷) to obtain 24. Example Input: 2, 4, 6, 8"*). The reasons of using this task are: (1) The reasoning process is highly transparent. For example, when we take 4 and 8 to achieve a 12 as "4 + 8 = 12", and then we would have 2, 6 from the original input set, and the newly generated 12 to continue to get 24, e.g., "2 × 6 + 12 = 24". The procedure can be naturally segmented into a sequence of steps; (2) The objective is clear, with well-defined evaluation criteria (whether the result equals 24 with all numbers and operations used correctly); and (3) No world knowledge or textual context is involved as the errors arise purely from the reasoning process, not from language or factual gaps. For each puzzle, the model generates an answer in eight reasoning steps, then we use `GPT-4o-mini` [OpenAI, 2024] to annotate whether it exhibits any of the three error types, as shown in Figure 3.

For each reasoning chain, we calculate its ACF and PE values with different rules for *abnormal* values: for ACF, we consider both abnormally low values (indicating inconsistent transition directions) and abnormally high values (often associated with rigid or repetitive progression), and for PE, we focus on significantly low values that reflect inefficient global trajectories. Specifically, we take the bottom or top 15.87% tails of the value distribution as abnormal following the 3σ rule under a Gaussian assumption. To quantify the association between stability metrics and logical errors, we perform significance tests for each error type and examine whether samples with abnormal metric values exhibit higher error occurrence rates. We employ both two-sample t-tests and *Mann–Whitney* tests to ensure our conclusions are robust under different distributional assumptions. We adopt a significance level of $\alpha = 0.05$, the null hypothesis assumes no association between stability metric anomalies and logical errors, and is rejected when $p < 0.05$.

As shown in Table 2, both ACF and PE exhibit statistically significant associations with logical errors across multiple settings. For semantic drift, both low and high ACF values are significantly associated with error occurrence. This phenomenon indicates that drift may arise from inconsistent local transitions or overly rigid progression. The association with PE is significant under the t-test but not under the *Mann–Whitney* test, which indicates a weaker effect. For the logical leap, all metrics show consistent significance across tests, which indicates that the logical leap is closely related to both local transition instability and global inefficiency. For repetition, high ACF and PE show a significant association. Overall, the null hypothesis of no association is rejected. The abnormal ACF and PE values are significantly associated with reasoning instability errors.

5.4 Ablation Study

We conduct ablation studies on GSM8K and GPQA using Qwen 2.5-1.5B-Instruct to analyze the individual contributions of ACF and PE reward, respectively. Specifically, we compare four variants: (1) Original *GRPO*; (2) Our StaRPO without PE reward (*StaRPO w/o PE*); (3) Our StaRPO without ACF reward (*StaRPO w/o ACF*); and (4) *StaRPO (Full)*, which contains both ACF and PE. For each method, we report both final result accuracy and process accuracy.

As shown in Figure 4, in terms of reasoning performance, both ACF and PE rewards contribute positively across two datasets.Removing either ACF or PE results in a noticeable decrease in both result and process accuracy, suggesting that each component captures a distinct aspect of reasoning quality for constraining reasoning stability during RL training. In particular, removing ACF results in a substantial drop in process accuracy, which indicates the importance of local transition consistency for maintaining structurally reliable reasoning paths. The unstable local semantic transitions can easily propagate and disrupt subsequent reasoning steps, even when the overall direction remains plausible. By contrast, combining ACF and PE consistently achieves the best performance across both metrics, indicating that local and global stability signals are complementary rather than redundant.

6 Conclusion

In this paper, we introduce StaRPO, a stability-augmented reinforcement policy optimization framework to enhance the logic stability of the LLM reasoning process. To capture the internal structure of reasoning trajectories, we focus on the notion of logical stability and decompose it into complementary local and global components. Two lightweight stability metrics, ACF and PE, are designed to capture step-wise consistency and goal-directed progression. Our experiments across four reasoning benchmarks demonstrate the effectiveness of StaRPO by comparing with existing strong baselines.

Acknowledgments

This work is supported in part by the National Science Foundation (NSF) under grants 2242812, 2550105, 2550106, 2543905, and 2616468.

References

Carl Anderson, Benjamin Vandenberg, Christopher Hauser, Alexander Johansson, and Nathaniel Galloway. Semantic coherence dynamics in large language models through layered syntax-aware memory retention mechanism. *Authroea: New York, NY, USA*, 2024.

Dibyanayan Bandyopadhyay, Soham Bhattacharjee, and Asif Ekbal. Thinking machines: A survey of llm based reasoning strategies. *arXiv:2503.10814*, 2025.

Jianghao Chen, Wei Sun, Qixiang Yin, Lingxing Kong, Zhixing Tan, and Jiajun Zhang. Ace-rl: Adaptive constraint-enhanced reward for long-form generation reinforcement learning. *arXiv:2509.04903*, 2025a.

Peter Chen, Xiaopeng Li, Ziniu Li, Wotao Yin, Xi Chen, and Tianyi Lin. Exploration vs exploitation: Rethinking rlvr through clipping, entropy, and spurious reward. *arXiv:2512.16912*, 2025b.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv:2503.09567*, 2025c.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4382. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/be91eb86eb74efc055cff83e953f86ce-Abstract-Conference.html.

Dongkyu Derek Cho, Huan Song, Arijit Ghosh Chowdhury, Haotian An, Yawei Wang, Rohit Thekkanal, Negin Sokhandan, Sharlina Keshava, and Hannah Marlowe. Breaking the safety-capability tradeoff: Reinforcement learning with verifiable rewards maintains safety guardrails in llms. *arXiv:2511.21050*, 2025.

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:2110.14168*, 2021.

Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. Using word embedding to evaluate the coherence of topics from twitter data. In *SIGIR*, 2016.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv:2505.10978*, 2025.

Christian George. The endorsement of the premises: Assumption-based or belief-based reasoning. *British Journal of Psychology*, 86(1):93–111, 1995.

Trudy Govier et al. *A practical study of argument*. Wadsworth Belmont, CA, 2010.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *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:2103.03874*, 2021.

Amin Hosseiny Marani and Eric PS Baumer. A review of stability in topic modeling: Metrics for assessing and techniques for improving stability. 2023.

Zijian Huang, Meng-Fen Chiang, and Wang-Chien Lee. Line: Logical query reasoning over hierarchical knowledge graphs. In *SIGKDD*, 2022.

Wendi Li and Yixuan Li. Process reward model with Q-value rankings. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/26494b66ae1114d314673e25b4967288-Abstract-Conference.html.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv:2503.22342*, 2025.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv:2412.19437*, 2024.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle. *arXiv:2509.16679*, 2025a.

Tongxuan Liu, Wenjiang Xu, Weizhe Huang, Yuting Zeng, Jiaxing Wang, Xingyu Wang, Hailong Yang, and Jing Li. Logic-of-thought: Injecting logic into contexts for full reasoning in large language models. In *NAACL*, 2025b.

Francois Luus, Prithviraj Sen, Pavan Kapanipathi, Ryan Riegel, Ndivhuwo Makondo, Thabang Lebese, and Alexander Gray. Logic embeddings for complex query answering. *arXiv:2103.00418*, 2021.

Math-AI. Aime 2024. https://huggingface.co/datasets/math-ai/aime24, 2024. Accessed: 2025-10-06.

Chau DM Nguyen, Tim French, Michael Stewart, Melinda Hodkiewicz, and Wei Liu. Representation learning in complex logical query answering on knowledge graphs: A survey. *ACM Computing Surveys*.

OpenAI. GPT-4o-mini: Advancing Cost-Efficient Intelligence, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2024-10-05.

Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. A survey of text representation and embedding techniques in nlp. *IEEe Access*, 11:36120–36146, 2023.

Federico Pennino, Bianca Raimondi, Massimo Rondelli, Andrea Gurioli, and Maurizio Gabbrielli. From reasoning to code: Grpo optimization for underrepresented languages. *arXiv:2506.11027*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

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

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/98711dea460bdefe0e651ca23ec98ba2-Abstract-Conference.html.

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:2402.03300*, 2024.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *ICCV*, 2023.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 2023.

Zhenwei Tang, Shichao Pei, Xi Peng, Fuzhen Zhuang, Xiangliang Zhang, and Robert Hoehndorf. Neural multi-hop logical query answering with concept-level answers. In *International Semantic Web Conference*, 2023.

Qwen Team. Qwen2 technical report. *arXiv:2407.10671*, 2, 2024.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Yining Wang, Jinman Zhao, Chuangxin Zhao, Shuhao Guan, Gerald Penn, and Shinan Liu. λ-grpo: Unifying the grpo frameworks with learnable token preferences. *arXiv:2510.06870*, 2025.

Junda Wu, Yuxin Xiong, Xintong Li, Zhengmian Hu, Tong Yu, Rui Wang, Xiang Chen, Jingbo Shang, and Julian McAuley. Ctrls: Chain-of-thought reasoning via latent state-transition. *arXiv:2507.08182*, 2025a.

Xingyu Wu, Yuchen Yan, Shangke Lyu, Linjuan Wu, Yiwen Qiu, Yongliang Shen, Weiming Lu, Jian Shao, Jun Xiao, and Yueting Zhuang. Lapo: Internalizing reasoning efficiency via length-adaptive policy optimization. *arXiv:2507.15758*, 2025b.

Jingyuan Yang, Dapeng Chen, Yajing Sun, Rongjun Li, Zhiyong Feng, and Wei Peng. Enhancing semantic consistency of large language models through model editing: An interpretability-oriented approach. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3343–3353, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Neurips*, 2023.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv:2506.18254*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv:2504.13837*, 2025.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv:2512.07783*, 2025a.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv:2509.02547*, 2025b.

Jinghan Zhang, Xiting Wang, Fengran Mo, Yeyang Zhou, Wanfu Gao, and Kunpeng Liu. Entropy-based exploration conduction for multi-step reasoning. *arXiv:2503.15848*, 2025c.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv:2509.08827*, 2025d.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv:2507.21848*, 2025e.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. Llm as a mastermind: A survey of strategic reasoning with large language models. *arXiv:2404.01230*, 2024.

Yi-Fan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, et al. R1-reward: Training multimodal reward model through stable reinforcement learning. *arXiv:2505.02835*, 2025f.

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

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 are include in the abstract and introduction.

   Guidelines:

   - The answer NA 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 NA 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: The limitations of the work are discussed in the experiment part.

   Guidelines:

   - The answer NA 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 provide the full set of assumptions and a complete (and correct) proof for all theoretical results.

Guidelines:

- The answer NA 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 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.

   Guidelines:

   - The answer NA 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 paper provide open access to the data used. The code will be release once accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specify all the training and test details necessary to understand the results.

   Guidelines:

   - The answer NA 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: The paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments.

   Guidelines:

   - The answer NA 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 NA 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 conductes in the paper conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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 discuss both potential positive societal impacts and negative societal impacts of the work in Introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA 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 nor research with human subjects.

    Guidelines:

    - The answer NA 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 nor research with human subjects.

    Guidelines:

    - The answer NA 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: [Yes]

    Justification: LLMs are central to both the method and evaluation setup: the paper trains and evaluates Qwen2.5-Instruct models and also uses GPT-4o-mini for annotating reasoning errors in Section 5.

    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.