

Know When to Reflect: Entropy-Guided Adaptive Self-Correction for LLM Reasoning

Xinglei Hao, Haiyu Zhao, Chunhong Zhang, Liping Liu, Jiarun Liu, Zheng Hu*

State Key Laboratory of Networking and Switching Technology,

Beijing University of Posts and Telecommunications

{haoxinglei, zhaohaiyu, zhangch, liuliping, liujiarun01, Huzheng}@bupt.edu.cn

Abstract

Self-correction has emerged as a promising paradigm for enhancing LLM reasoning, yet recent studies reveal a troubling phenomenon: without external oracle feedback, iterative reflection often degrades rather than improves performance. Our analysis uncovers that *toxic reflection*, where correct answers deteriorate into incorrect ones, occurs at rates comparable to effective corrections. We identify the root cause as the *shortsightedness* of existing methods: models focus only on the previous output while neglecting the global reasoning state, rendering them unable to quantify uncertainty. To address this, we propose two complementary reflection signals—**Reflecting Entropy** for snapshot disagreement among reasoning trajectories and **Belief Shift** for inter-round belief revision—paired with a two-stage hierarchical controller that schedules reflection adaptively. We instantiate these ideas as **REGAR** (Reflecting Entropy-Guided Adaptive Reflection), a framework that maintains a pool of candidate reasoning trajectories and applies a single, uniform reflection primitive when correction is needed. Experiments on mathematical and commonsense reasoning demonstrate consistent improvements over strong baselines while reducing computational overhead, highlighting both effectiveness and efficiency. Our code is available at <https://github.com/apricitytt/REGAR>.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various natural language processing tasks (Gou et al., 2023; OpenAI et al., 2024; Touvron et al., 2023; Zhang et al., 2022), particularly in complex reasoning scenarios. To unlock the reasoning potential within LLMs, researchers have proposed a series of prompting strategies such as Chain-of-Thought (Wei et al., 2022), Tree of Thoughts (Yao

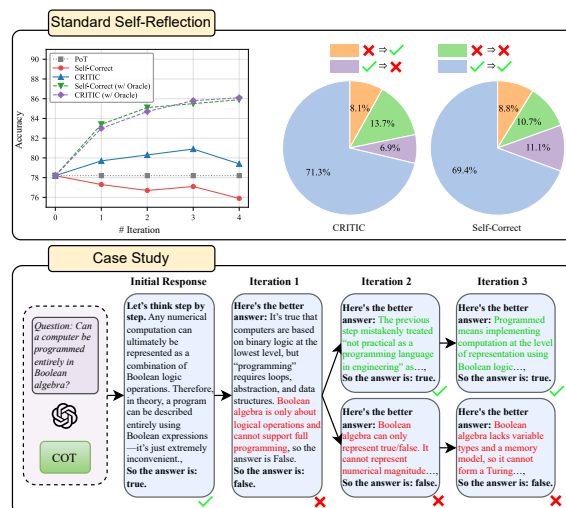


Figure 1: (Top) Iterative reflection on GSM8K with GPT-3.5-turbo, reproducing the setting of Huang et al. (2023) that originally motivated the toxic-reflection problem; we retain this setup for direct comparability with the literature, and reproduce the same near-parity in-distribution on the models and benchmarks of our own evaluation in §4.4. (Bottom) Schematic of cognitive dissonance arising under repeated reflection on the same input.

et al., 2023), and other prompting techniques (Yao et al., 2022; Zhou et al., 2022; Wang et al., 2022) to further enhance reasoning performance. Inspired by how humans iteratively refine their thinking (Madaan et al., 2023), the concept of *self-correction* has emerged as a promising direction, where LLMs refine their responses based on feedback to their previous responses (Welleck et al., 2022; Shinn et al., 2023; Paul et al., 2024). This paradigm has garnered extensive attention and appears to endow LLMs with the potential to autonomously improve their reasoning quality.

However, recent studies have indicated that models often exhibit cognitive dissonance when faced with simple self-correction prompts (Huang et al., 2023; Valmeekam et al., 2023; Zhang et al., 2024).

*Corresponding author.

During reflection, these models display polarized self-correction behaviors, either clinging to incorrect answers or producing inconsistent responses after multiple reflections. Tyen et al. (2024) found that while LLMs are not entirely incapable of detecting reasoning errors, they exhibit significant deficiencies in *error detection*. Recent work on the “Self-Correction Blind Spot” (Tsui, 2025) reveals a systematic problem: LLMs can successfully identify and correct external errors yet remain oblivious to errors in their own reasoning. Another major obstacle stems from the model’s *sycophantic* nature (Wei et al., 2024; Zhang et al., 2025; Wang et al., 2025), which not only prevents the model from upholding the truth but may also cause it to actively modify correct answers when confronted with doubt in the prompt. Based on the aforementioned research, we conducted experimental analysis. As shown in Figure 1 (top), the left part, following Huang et al. (2023), introduces an oracle label—a perfect external validator that guides the model on when to reflect—to verify whether LLMs possess the intrinsic capability to improve through reflection. With oracle guidance, reflection steadily improves accuracy; without it, performance fluctuates and even degrades, demonstrating that models lack reliable self-assessment capabilities. The right part further quantifies this issue: the proportion of effective reflections ($\times \rightarrow \checkmark$) is nearly equal to that of toxic reflections ($\checkmark \rightarrow \times$), confirming that reflection without uncertainty awareness becomes essentially random.

As shown in Figure 1 (bottom), we find that an important reason for cognitive dissonance during the reflection process is the shortsightedness of the model, which exacerbates uncertainty in the reasoning process. In the multi-round iterative reflection process, when the model reflects at step t , it typically focuses only on the output from the previous round (step $t - 1$) and does not adequately consider the reasoning states from all prior steps and their impact on the current reasoning path. This local perspective in the reflection strategy makes it difficult for the model to form long-term reasoning plans, resulting in adjustments that, while seemingly corrective in each round, may actually lead it further away from the correct solution.

To address this limitation, we first introduce **Reflecting Entropy**, a process-level metric inspired by Semantic Entropy (Farquhar et al., 2024) and designed to quantify the uncertainty of an evolving reasoning process. Specifically, Reflecting En-

tropy groups reasoning trajectories into semantic clusters at multiple reasoning depths, computes each cluster’s probability mass using confidence weights, and aggregates the resulting Shannon entropies into a single snapshot disagreement signal. Snapshot disagreement alone, however, cannot tell whether an apparent consensus has truly settled or has just been pushed into agreement and is still mid-revision; we therefore introduce a complementary signal, **Belief Shift**, which measures how much the cluster distribution has moved between consecutive rounds. Built upon these two signals, we propose **REGAR** (Reflecting Entropy-Guided Adaptive Reflection), a framework that employs a two-stage hierarchical controller: high Reflecting Entropy directly triggers reflection, while low Reflecting Entropy hands the decision to Belief Shift—a stable belief distribution leads to termination, whereas an unstable one triggers another round of reflection.

Our contributions can be summarized as follows:

- Our analysis reveals that static reflection pipelines suffer from toxic reflection due to the inability to quantify uncertainty during reasoning.
- We propose two complementary reflection signals—*Reflecting Entropy* for snapshot disagreement and *Belief Shift* for inter-round revision—paired with a two-stage hierarchical controller that schedules reflection adaptively.
- We instantiate these ideas as the **REGAR** framework and demonstrate consistent effectiveness and efficiency gains across five contemporary LLMs and four reasoning benchmarks.

2 Related Work

2.1 Evolution of Self-Correction Mechanisms

Self-correction in LLMs has evolved from intrinsic introspection to structured agentic frameworks. Early approaches such as SELF-REFINE (Madaan et al., 2023) and REFLEXION (Shinn et al., 2023) demonstrated that LLMs can iteratively refine outputs through self-generated feedback, achieving promising results in code generation and mathematical reasoning. Subsequent work enhanced robustness by introducing external tools for verification (Gou et al., 2023) or multi-agent debate to mitigate individual cognitive bias (Liang et al., 2024; Du et al., 2024).

Despite these advances, intrinsic self-correction remains unreliable. Recent studies show that without oracle feedback, self-correction often leads to unstable behavior or performance degradation (Huang et al., 2023; Valmeekam et al., 2023). To address this, recent efforts explore higher-level control mechanisms: Liu et al. (2025) introduce meta-thinking to guide models toward abstract logical structures, while Ma et al. (2025) leverage reinforcement learning to reward effective corrections.

2.2 Cognitive Dissonance in LLM Self-Reflection

Beyond mechanism design, recent work examines self-correction failures from a cognitive perspective. Tyen et al. (2024) identify a fundamental asymmetry: models struggle with *error detection* but exhibit strong *error correction* once errors are externally identified. This phenomenon is formalized as the “Self-Correction Blind Spot” by Tsui (2025), showing that models can correct errors in others’ reasoning while remaining unaware of their own flaws. Compounding this, sycophantic tendencies lead models to revise correct answers when exposed to doubt or conflicting feedback (Wei et al., 2024; Zhang et al., 2025; Wang et al., 2025). Together, these findings highlight cognitive dissonance: models lack the ability to assess their own reasoning reliability, causing reflection to become ineffective or harmful.

Recent research incorporates uncertainty signals to mitigate such issues. Wu et al. (2025) introduce Semantic Entropy into Chain-of-Thought optimization, Chen et al. (2026) use confidence signals for early exits in debate, and Luo et al. (2025) propose logical symmetry as a reliability proxy. All of these signals are, however, *single-snapshot and endpoint-only* by construction: by scoring only the current answer, they neither see disagreement that arises early and later collapses into surface agreement, nor tell whether an apparent consensus is still being rewritten across rounds. We re-implement the two most comparable methods under our protocol, and both trail REGAR by a clear margin (§4.4). Our work addresses the gap with two complementary signals—*Reflecting Entropy* for snapshot disagreement and *Belief Shift* for inter-round revision—enabling models to perceive both the current state and the dynamics of their own reasoning.

3 Methodology

We propose **REGAR** (Reflecting Entropy-Guided Adaptive Reflection), a framework that quantifies the uncertainty of LLM reasoning through two complementary signals—a primary *Reflecting Entropy* that captures snapshot disagreement among trajectories and an auxiliary *Belief Shift* that tracks inter-round belief revision—and uses them in a hierarchical controller to decide adaptively when to reflect. The framework consists of three components, illustrated in Figure 2: (i) multi-perspective initialization populates a weighted trajectory pool (§3.1); (ii) Reflecting Entropy $\tilde{H}^{(t)}$ and Belief Shift $\Delta^{(t)}$ are computed from the pool (§3.2); and (iii) a hierarchical controller decides between reflection and termination (§3.3). When reflection is invoked, an LLM arbiter analyzes the two leading reasoning paths and produces a refined trajectory; feedback from each round updates trajectory weights via a Linear Reward–Penalty rule, creating a closed-loop search in semantic space.

3.1 Problem Formulation and Initialization

Trajectory Pool Formulation. Given an input query x , our goal is to track the system’s evolving belief over candidate answers by maintaining a trajectory pool at iteration t :

$$\text{TP}^{(t)} = \{(y_i, w_i^{(t)})\}_{i=1}^N, \quad (1)$$

where $t \in \{0, 1, \dots, T_{\max}\}$ denotes the current reflection iteration, $y_i = (s_i^1, s_i^2, \dots, s_i^{L_i})$ is the i -th reasoning trajectory expressed as an ordered sequence of reasoning steps (e.g., code statements in Program-of-Thought, or sentences in Chain-of-Thought), and $w_i^{(t)} \in [0, 1]$ is its *heuristic confidence weight*, normalized so that $\sum_i w_i^{(t)} = 1$ and used as relative belief-mass bookkeeping over candidate trajectories. Structured feedback produced during reflection updates these weights over rounds (§3.3) through the Linear Reward–Penalty rule. We emphasize that this update is a heuristic asymmetric reinforcement rule rather than a Bayesian posterior update; $w_i^{(t)}$ should accordingly be read as a bookkeeping quantity that the clustering and controller consume, not as an estimate of any true posterior.

Perspective Generation. To populate the pool with genuinely diverse trajectories rather than redundant samples, given a query x , the LLM generates a set of distinct prompt instructions $\mathcal{I} = \{I_1, I_2, \dots, I_m\}$, each corresponding to a special-

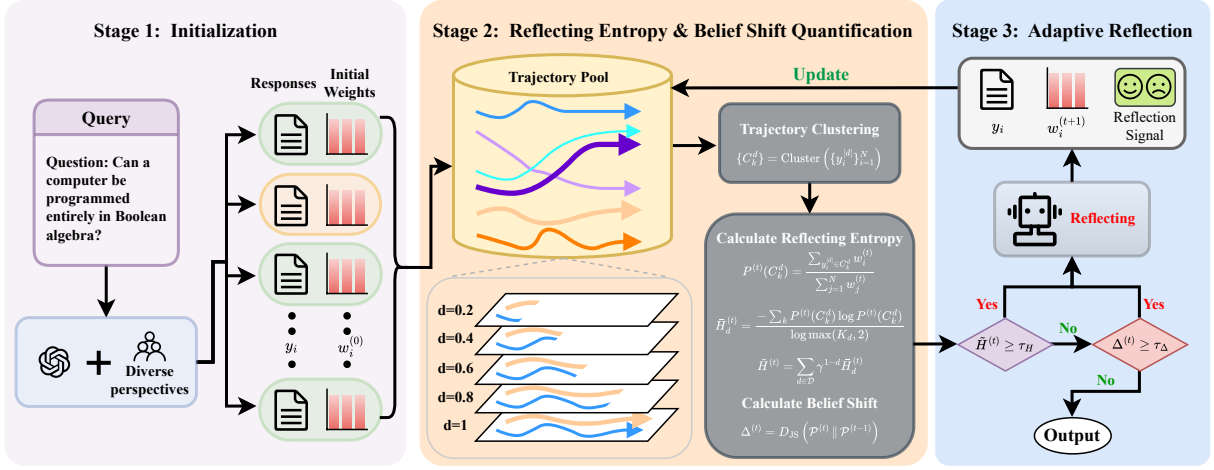


Figure 2: The REGAR framework. Multi-perspective initialization populates a weighted trajectory pool; Reflecting Entropy $\tilde{H}^{(t)}$ and Belief Shift $\Delta^{(t)}$ are computed from cluster distributions and feed a two-stage hierarchical controller that issues either REFLECT or TERMINATE.

ized reasoning perspective (e.g., checking boundary conditions or constructing counterexamples). Both the number of perspectives m and their content are decided by the model, enabling flexible allocation of cognitive resources.

Uniform Initialization. For each instruction I_k , we sample a reasoning trajectory y_k with moderate temperature to encourage diversity. We assign equal initial confidence to all trajectories: $w_k^{(0)} = 1/m$. This uniform initialization ensures that all reasoning perspectives receive fair consideration at the outset, allowing the subsequent reflection process—rather than prior probability biases—to determine trajectory quality through evidence accumulation. After initialization, the m trajectories form the initial pool $TP^{(0)}$, and we henceforth index trajectories uniformly by i .

3.2 Reflecting Entropy and Belief Shift

Reflection control needs two distinct signals: a snapshot uncertainty over the current pool, and a measure of how much the belief is moving between rounds. We call them **Reflecting Entropy** (snapshot disagreement) and **Belief Shift** (inter-round movement), and treat the former as the primary control signal and the latter as an auxiliary safety check.

Trajectory Clustering at Multiple Depths. Both signals require comparing trajectories at the *belief* level rather than the *surface* level: two trajectories that arrive at the same conclusion via different wording should contribute to the same belief mass, while genuinely divergent reasoning routes should remain separable. We therefore group tra-

jectories into semantic clusters $\{C_k^d\}$, where the superscript $d \in [0, 1]$ is a normalized prefix depth indicating that the cluster is formed over the depth- d prefixes $y_i^{[d]} = (s_i^1, \dots, s_i^{[d \cdot L_i]})$. Computing clusters at multiple depths lets Reflecting Entropy capture not only final-answer disagreement ($d=1.0$) but also early-stage divergence among reasoning processes that may later collapse into apparent consensus. We use $\mathcal{D} = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, an evenly spaced grid that empirically suffices to characterize the trajectory’s evolution from initial setup to final answer. Clustering itself is performed by a three-level equivalence cascade—symbolic, rule-based, and a lightweight equivalence judge—with implementation details deferred to Appendix B.

Reflecting Entropy (primary signal). Given the depth- d clusters $\{C_k^d\}$ produced above, the cluster probability under the current weights is

$$P^{(t)}(C_k^d) = \frac{\sum_{y_i^{[d]} \in C_k^d} w_i^{(t)}}{\sum_{j=1}^N w_j^{(t)}}, \quad (2)$$

and the depth- d Shannon entropy is

$$H_d^{(t)} = - \sum_k P^{(t)}(C_k^d) \log P^{(t)}(C_k^d), \quad (3)$$

which is zero when all trajectories agree at depth d and grows as belief mass spreads over competing clusters. For cross-query comparability we normalize $H_d^{(t)}$ by its query-dependent upper bound:

$$\tilde{H}_d^{(t)} = \frac{H_d^{(t)}}{\log \max(K_d, 2)} \in [0, 1], \quad (4)$$

where $K_d = |\{C_k^d\}|$. We then aggregate across depths via a discount-weighted sum:

$$\tilde{H}^{(t)} = \sum_{d \in \mathcal{D}} \gamma^{1-d} \cdot \bar{H}_d^{(t)}, \quad (5)$$

where $\gamma \in (0, 1]$ is a depth discount that grants the largest weight to the endpoint depth $d = 1.0$ ($\gamma^0 = 1$) while still admitting non-trivial signal from earlier depths, so that disagreement about the answer outweighs disagreement about the route taken to reach it.

Belief Shift (auxiliary signal). Reflecting Entropy describes a single snapshot, but a snapshot alone cannot distinguish a pool that has genuinely settled from one that has just been pushed into apparent agreement and is still mid-revision. We therefore introduce a second signal $\Delta^{(t)}$, defined as the Jensen–Shannon divergence (Lin, 1991) between the endpoint cluster distributions of two consecutive rounds, with the depth restricted to $d=1.0$ to filter out superficial step-level rephrasing at intermediate depths. Concretely, let

$$\mathcal{P}^{(t)} = (P^{(t)}(C_1^{1.0}), P^{(t)}(C_2^{1.0}), \dots) \quad (6)$$

denote the round- t endpoint cluster probability vector, where each component is given by Eq. 2 with $d=1.0$. The Belief Shift is then

$$\Delta^{(t)} = D_{\text{JS}}(\mathcal{P}^{(t)} \parallel \mathcal{P}^{(t-1)}) \in [0, \log 2], \quad (7)$$

defined for $t \geq 1$, with $\Delta^{(0)} \triangleq 0$ by convention since the pool has not yet evolved at the initial round. Intuitively, $\Delta^{(t)}$ measures how much belief mass reflection has just rewritten: a small $\Delta^{(t)}$ indicates that the round- t belief distribution is essentially the same as the round- $(t-1)$ one—the pool has stopped moving—while a large $\Delta^{(t)}$ signals that reflection has just relocated a substantial portion of the belief mass. The two signals are thus complementary: $\tilde{H}^{(t)}$ asks whether the pool is fragmented, $\Delta^{(t)}$ whether it is still moving. All symbols are summarized in Appendix A, and Appendix H instantiates every quantity numerically on worked examples.

3.3 Hierarchical Adaptive Reflection

At round $t=0$ the pool has not yet evolved, so $\Delta^{(0)} \triangleq 0$ and the decision (REFLECT or TERMINATE) is effectively governed by $\tilde{H}^{(0)}$ alone: reflection is triggered whenever the initial pool is fragmented *at any depth*, and the controller terminates without invoking the arbiter only when the

trajectories agree throughout. From $t \geq 1$ both signals are available, and the two-stage rule below applies uniformly.

Two-Stage Hierarchical Decision. With two thresholds $\tau_H, \tau_\Delta > 0$, the controller acts according to

$$\text{Action}^{(t)} = \begin{cases} \text{REFLECT}, & \tilde{H}^{(t)} \geq \tau_H \\ \text{TERMINATE}, & \tilde{H}^{(t)} < \tau_H, \Delta^{(t)} < \tau_\Delta \\ \text{REFLECT}, & \tilde{H}^{(t)} < \tau_H, \Delta^{(t)} \geq \tau_\Delta \end{cases} \quad (8)$$

The two stages address distinct failure modes. **Stage 1** uses Reflecting Entropy alone: if $\tilde{H}^{(t)} \geq \tau_H$ the pool is *fragmented*, and reflection proceeds. **Stage 2** is reached when $\tilde{H}^{(t)} < \tau_H$, i.e., the pool appears to have converged; here Belief Shift acts as a stability check. An apparently converged snapshot with $\Delta^{(t)} < \tau_\Delta$ is genuinely *stable*, and the controller terminates. An apparently converged snapshot with $\Delta^{(t)} \geq \tau_\Delta$ is *unstable*: the belief mass is still being rewritten across rounds, indicating a *pseudo-consensus* rather than true agreement.

Reflection Step. When reflection is invoked, the controller takes the two endpoint clusters C_A, C_B with the largest cumulative weights and forwards their highest-weight representatives y_A, y_B to an LLM arbiter (the same model used to generate trajectories, keeping the framework strictly self-correcting). The arbiter is asked to (i) identify the reasoning errors in each of y_A and y_B , and (ii) emit a single refined trajectory y_{new} . The arbiter is not required to side with either input: y_{new} may endorse y_A , endorse y_B , or introduce a third answer that disagrees with both.

The arbiter’s verdict yields feedback labels $\mathcal{S}^\pm$: trajectories with identified errors receive $\mathcal{S}^-$, and the side endorsed by y_{new} (or y_{new} itself if it is a third answer) receives $\mathcal{S}^+$. These labels drive the weight update below.

Pool Update. After each reflection round, y_{new} is appended to the pool with initial weight

$$w_{\text{new}}^{(t+1)} = \frac{1}{|\text{TP}^{(t)}| + 1}, \quad (9)$$

and all existing weights are renormalized to sum to one. Existing trajectories are never overwritten or deleted; they can only be down-weighted by the LRP update below, or pruned when $w_i < \epsilon$.

Weight Update via Linear Reward–Penalty. The $\mathcal{S}^+/\mathcal{S}^-$ tags propagate from y_A, y_B to all trajectories in their respective endpoint clusters C_A, C_B (i.e., $\{C_k^{1.0}\}$): every $y_i \in C_A$ inherits whichever tag was assigned to y_A , and likewise

for C_B ; y_{new} receives $\mathcal{S}^+$ on its own. Cluster-level propagation reflects the granularity of the arbiter’s verdict, which is defined over final answers and therefore over endpoint clusters. The confidence weights are then updated via the Linear Reward–Penalty rule (Narendra and Thathachar, 2012), a classical update scheme that asymmetrically rewards positive feedback and discounts negative feedback:

$$w_i^{(t+1)} \leftarrow \begin{cases} w_i^{(t)} + \alpha(1 - w_i^{(t)}), & \text{if } \mathcal{S}^+ \\ w_i^{(t)} \cdot \rho, & \text{if } \mathcal{S}^- \end{cases} \quad (10)$$

where $\alpha \in (0, 1)$ is the reinforcement rate and $\rho \in (0, 1)$ is the decay factor. The asymmetry is deliberate: an endorsed trajectory moves a fraction α of the remaining distance toward weight 1, whereas a refuted one is shrunk multiplicatively, so evidence accumulates gradually while discredited routes lose influence quickly. After local updates, weights are globally renormalized:

$$w_i^{(t+1)} \leftarrow \frac{w_i^{(t+1)}}{\sum_{j=1}^N w_j^{(t+1)}}. \quad (11)$$

Termination. Reflection terminates when either (i) the two-stage rule in Eq. 8 returns TERMINATE, indicating that the pool has reached a stable consensus, or (ii) the iteration budget T_{max} is exhausted. The final prediction is the highest-weight trajectory in the dominant cluster of the terminal pool.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate our method across two representative reasoning scenarios: *mathematical reasoning* and *commonsense reasoning*. For mathematical reasoning, we adopt MATH-500 (Lightman et al., 2024), AIME (Mathematical Association of America, 2024), and GSM8K (Cobbe et al., 2021). For commonsense reasoning, we use StrategyQA (Geva et al., 2021). MATH-500 and AIME serve as the main difficulty tier alongside StrategyQA; GSM8K is retained as a saturated easy-task control and its results are reported in Appendix E. Detailed statistics and descriptions of these datasets are provided in Appendix C.2.

Implementation Details. Experiments are conducted using five contemporary LLMs covering open-source and proprietary families: Qwen3-8B and Qwen3-32B (Yang et al., 2025), Llama3.1-70B-Instruct (Grattafiori et al., 2024), DeepSeek-V3 (Liu et al., 2024), and GPT-4o (OpenAI et al.,

2024). Unless otherwise specified, the temperature is fixed at 0.3 and the iteration budget is $T_{\text{max}} = 4$. The controller thresholds are fixed at $\tau_H = 0.55$ and $\tau_\Delta = 0.10$ for all models and benchmarks; the depth discount is $\gamma = 0.7$ and the number of perspectives lies in $m \in [2, 6]$. These values are selected once on a held-out development split of AIME that is disjoint from all evaluation sets, then frozen and held constant throughout all experiments; sensitivity analyses confirming their robustness are reported in Appendix C.3.1. To reduce variance, we perform five independent runs on each dataset and report the average performance as the final result. All methods are evaluated under one uniform and deliberately conservative protocol—non-thinking mode for the hybrid Qwen3 models, zero-shot prompting, single low-temperature decoding, and strict automatic answer matching—which is specified in full in Appendix C.1. Detailed hyperparameter settings and prompt templates are provided in Appendix C.3 and D, respectively.

4.2 Baselines

We compare the proposed REGAR framework against a diverse set of representative baselines, organized into four categories. *Non-reflective*: (1) Chain-of-Thought (CoT) (Wei et al., 2022); (2) Program-of-Thought (PoT) (Chen et al., 2023). *Reflective with external feedback*: (3) CRITIC (Gou et al., 2023), which uses tool-interactive critiquing. *Sampling- and ensemble-based reflective methods*: (4) Self-Consistency (SC) at $k=8$ (Wang et al., 2022), and a *budget-matched* variant (SC-BM) constrained to the same total token budget as REGAR; (5) Multi-Agent Debate (Du et al., 2024; Liang et al., 2024); (6) Self-Contrast (Zhang et al., 2024); (7) Instruct-of-Reflection (IoRT) (Liu et al., 2025) and a budget-matched variant (IoRT-BM). *Uncertainty-guided*: (8) SE-CoT (Wu et al., 2025), which halts re-sampling once the sampled answers reach a low-entropy consensus; (9) RES (Luo et al., 2025), which estimates answer reliability from reasoning–explanation symmetry without iterative reflection. The two budget-matched baselines are designed to isolate accuracy gains attributable to the algorithm itself rather than to differences in raw computational allocation, directly addressing concerns that token-heavier methods could inflate apparent improvements. As neither uncertainty-guided method reports results under our protocol, we re-implement both ourselves; implementation details are given in Appendix F.

Methods	MATH-500					AIME					#Tokens
	Q3-8B	Q3-32B	L3-70B	DS-V3	GPT-4o	Q3-8B	Q3-32B	L3-70B	DS-V3	GPT-4o	Avg.
CoT	56.8	80.5	67.3	78.4	75.8	8.3	18.5	12.2	16.5	13.3	493
PoT	60.2	82.1	70.5	80.2	78.0	10.0	20.4	14.5	18.7	16.7	471
CRITIC	58.5	81.0	69.0	79.3	76.5	9.2	19.6	13.3	17.6	15.0	3127
Self-Consistency ($k=8$)	62.5	84.3	72.8	82.6	80.5	13.3	25.5	18.3	22.4	22.5	4063
SC (budget-matched)	60.8	83.0	70.8	81.0	79.0	11.7	22.8	16.5	19.8	19.6	2872
Multi-Agent Debate	61.0	83.2	71.5	81.5	79.5	11.7	23.2	17.0	20.5	20.0	3946
Self-Contrast	61.8	83.8	72.0	81.8	80.0	12.5	24.0	17.4	21.0	20.8	3554
IoRT	63.0	84.6	73.2	82.8	80.8	14.2	25.8	18.8	22.6	23.0	3827
IoRT (budget-matched)	61.5	83.5	71.5	81.3	79.5	12.5	23.5	17.0	20.0	20.6	2864
SE-CoT	61.9	83.8	71.8	81.6	79.8	12.7	23.8	17.3	20.4	21.0	2653
RES	60.9	82.8	71.0	80.6	78.5	12.0	22.6	16.4	19.5	19.8	982
REGAR (Ours)	65.2	86.8	75.6	84.7	83.0	17.5	29.4	22.5	26.2	27.0	2843

Table 1: Final-answer accuracy (%) on MATH-500 and AIME across five LLMs. Q3-: Qwen3; L3-70B: Llama3.1-70B-Instruct; DS-V3: DeepSeek-V3. #Tokens Avg. is the average *total* token consumption per problem, counting all pipeline stages and both input and output tokens, averaged over five runs (see Appendix G). “BM” denotes baselines restricted to REGAR’s token budget. REGAR uses PoT for initialization. SE-CoT and RES are re-implemented by us under the identical protocol (Appendix F). **Bold**: best accuracy per column; in the #Tokens column, **bold** marks the lowest consumption among *reflective* methods. GSM8K results: Appendix E.

4.3 Main Results

Mathematical Reasoning. Table 1 summarizes the performance of REGAR against eleven representative baselines on MATH-500 and AIME. Across all five LLMs and both benchmarks, REGAR achieves the best final-answer accuracy. Compared to the non-reflective PoT baseline, REGAR yields an average gain of **+4.9%** on MATH-500 and **+8.5%** on AIME; compared to the strongest reflective baseline IoRT at full budget, the corresponding gains are **+2.2%** and **+3.6%**, while REGAR consumes **26% fewer tokens** (2843 vs. 3827). Under matched token budgets (SC-BM, IoRT-BM), REGAR retains a 3.3–4.8% advantage on MATH-500 and 5.0–7.4% on AIME. The gap widens with task difficulty: AIME shows the largest gains across every model column, while GSM8K (Appendix E) leaves only 0.2–0.4% room for further improvement.

Commonsense Reasoning. Table 2 compares REGAR against representative baselines on StrategyQA across the same five LLMs. REGAR consistently outperforms all baselines, achieving an average gain of **+5.5%** over CoT and **+2.6%** over IoRT, while the budget-matched variants again fall below REGAR. Token consumption is reduced by **30%** relative to Self-Consistency (2812 vs. 4061) and **26%** relative to IoRT (2812 vs. 3819), confirming that the two-signal framework generalizes beyond mathematical reasoning with both effec-

Methods	Q3-8B	Q3-32B	L3-70B	DS-V3	GPT-4o	Tokens
CoT	70.5	76.8	78.5	77.2	80.2	524
CRITIC	70.0	76.2	78.0	76.8	79.8	3018
Self-Consistency ($k=8$)	73.2	79.5	81.3	80.0	82.0	4061
SC (budget-matched)	71.8	78.2	80.0	78.5	81.0	2839
Multi-Agent Debate	72.0	78.5	80.5	78.8	81.3	3878
Self-Contrast	72.5	79.0	80.8	79.3	81.5	3492
IoRT	73.5	79.8	81.5	80.3	82.5	3819
IoRT (budget-matched)	72.0	78.5	80.2	78.7	81.2	2841
REGAR (Ours)	76.0	82.5	84.0	83.0	85.0	2812

Table 2: Final-answer accuracy (%) and #Tokens on StrategyQA across five LLMs. REGAR uses CoT for initialization. “BM” restricts each baseline to REGAR’s token budget. **Bold**: best accuracy per column; in the Tokens column, the lowest among *reflective* methods.

tiveness and efficiency benefits.

4.4 Does REGAR Actually Reduce Toxic Reflection?

Final accuracy alone cannot tell whether REGAR wins by suppressing toxic reflection or merely by the better starting point that multi-perspective initialization provides. We therefore measure per-instance transitions directly on Qwen3-8B, against standard iterative self-reflection (Huang et al., 2023) under a matched iteration budget.

Table 3 supports two conclusions. First, **toxic transitions occur at essentially the same rate as effective corrections** even on a strong contemporary model (AIME: 4.9% vs. 5.8%; MATH-500: 4.2% vs. 5.2%), so standard self-reflection buys only a marginal net gain (+0.9%/+1.0%)—

Method	Data	Init.	Toxic ($\checkmark \rightarrow \times$)	Effect. ($\times \rightarrow \checkmark$)	Net	Final
Std. Self-Reflect.	AIME	10.0	4.9	5.8	+0.9	10.9
REGAR	AIME	13.0	1.5	6.0	+4.5	17.5
Std. Self-Reflect.	MATH	60.2	4.2	5.2	+1.0	61.2
REGAR	MATH	61.5	1.3	5.0	+3.7	65.2

Table 3: Per-instance answer-transition rates (%) on Qwen3-8B. **Init.**: accuracy before reflection; **Net** = Effect. – Toxic. MATH: MATH-500.

reproducing *in-distribution* the near-parity that Figure 1 shows on GPT-3.5-turbo/GSM8K. Second, **REGAR suppresses toxic transitions by roughly $3\times$ ($4.9\% \rightarrow 1.5\%$, $4.2\% \rightarrow 1.3\%$) while preserving effective corrections**, so its net gain is several times larger. The w/o Persp. ablation (Table 4) rules out the alternative explanation, since removing multi-perspective initialization costs only 2.7% (MATH-500) / 3.0% (AIME) on this same model, while the controller—and hence the suppressed toxic rate—remains.

Finally, we ask whether existing uncertainty signals already achieve this. Re-implementing the two closest methods under the identical protocol across all five LLMs (Table 1), SE-CoT and RES trail REGAR by **3.0–4.6%** on MATH-500 and **4.8–7.2%** on AIME on every model column (e.g., on Qwen3-8B, SE-CoT reaches 61.9/12.7 and RES 60.9/12.0 vs. REGAR’s 65.2/17.5). Both fall short for the same structural reason—their signals are answer-level and single-round, and therefore cannot represent pseudo-consensus—as detailed in Appendix F.

4.5 Ablation Study

We perform an ablation study to assess the contribution of each core component in the REGAR framework:

- **REGAR (w/o Persp.)**: Multi-perspective initialization is removed; instead of generating diverse trajectories from different perspectives, we use a single perspective and sample multiple trajectories with temperature-based randomness.
- **REGAR (w/o Multi-Depth)**: The multi-depth aggregation in Reflecting Entropy is disabled by restricting $\mathcal{D} = \{1.0\}$, so $\tilde{H}^{(t)}$ reduces to a single snapshot at the endpoint depth.
- **REGAR (w/o Belief Shift)**: The Stage-2 stability check is disabled by dropping $\Delta^{(t)}$ from

Model	Dataset	Initial	w/o Persp.	w/o Multi-Depth	w/o Belief Shift	REGAR
Q3-8B	MATH-500	60.2	62.5	64.0	63.5	65.2
	AIME	10.0	14.5	16.0	14.5	17.5
	StrategyQA	70.5	73.0	74.5	73.5	76.0
	GSM8K	89.2	89.7	90.5	90.6	91.0
Q3-32B	MATH-500	82.1	84.8	85.7	85.2	86.8
	AIME	20.4	26.5	27.5	25.4	29.4
	StrategyQA	76.8	80.0	81.0	80.5	82.5
	GSM8K	94.0	94.0	94.6	94.7	95.2
L3-70B	MATH-500	70.5	73.0	74.3	74.0	75.6
	AIME	14.5	19.8	21.0	19.5	22.5
	StrategyQA	78.5	81.5	82.5	82.0	84.0
	GSM8K	94.8	95.0	95.7	95.5	96.0
DS-V3	MATH-500	80.2	82.5	83.5	83.0	84.7
	AIME	18.7	23.5	24.5	23.0	26.2
	StrategyQA	77.2	80.5	81.5	81.0	83.0
	GSM8K	94.2	94.4	95.0	95.0	95.5
GPT-4o	MATH-500	78.0	80.8	81.8	81.5	83.0
	AIME	16.7	24.5	25.5	24.0	27.0
	StrategyQA	80.2	82.5	83.5	83.0	85.0
	GSM8K	94.9	95.3	95.7	95.5	95.9

Table 4: Component ablation across five LLMs and four benchmarks (accuracy %). **Initial** is the non-reflective baseline (PoT for math, CoT for StrategyQA); the three middle columns drop one REGAR component each; **REGAR** is the full method.

the controller; termination is committed whenever $\tilde{H}^{(t)} < \tau_H$.

Multi-perspective initialization expands the solution space. REGAR (w/o Persp.) drops by an average of **2.3%** on MATH-500 and **2.8%** on AIME, with the largest losses on weaker open-source models (Q3-8B, L3-70B). This confirms that diversity from explicit perspective prompts—rather than from sampling variance alone—is essential when the underlying task admits multiple legitimate solution routes.

Multi-depth aggregation adds depth-resolved signal. REGAR (w/o Multi-Depth) costs **1.2%** on MATH-500 and **1.6%** on AIME on average. The modest gap reflects that the endpoint depth already captures the dominant disagreement; the consistent positive contribution from earlier depths nonetheless confirms that intermediate reasoning states carry information beyond the final answer.

Belief Shift is the largest single contributor on hard tasks. REGAR (w/o Belief Shift) causes the largest drop on AIME—an average of **3.2%**, peaking at **4.0%** on Q3-32B ($29.4\% \rightarrow 25.4\%$). This is the most direct empirical evidence that inter-round belief revision is a non-redundant signal: without $\Delta^{(t)}$ the controller cannot distinguish stable convergence from pseudo-consensus and terminates prematurely on a non-trivial fraction of hard instances. The gap shrinks to 1.6% on MATH-500 and nearly vanishes on the saturated GSM8K, consistent with pseudo-consensus arising primarily on genuinely hard problems.

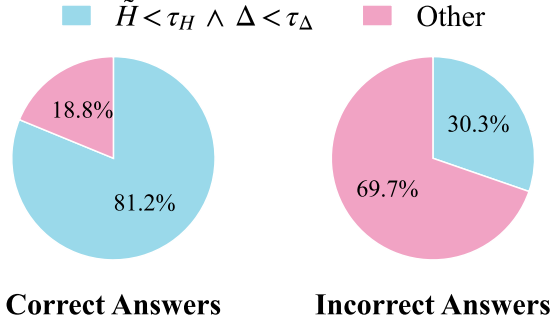


Figure 3: Terminal controller states on MATH-500 (Qwen3-32B), split by correctness. *Stable*: $\tilde{H}^{(t^*)} < \tau_H \wedge \Delta^{(t^*)} < \tau_\Delta$; *Other*: the converse. 81.2% of correct answers are stable; 69.7% of incorrect answers are not.

Hyperparameter sensitivity analyses confirming that all defaults sit on broad performance plateaus are deferred to Appendix C.3.1.

5 Discussions

Why the Two-Signal Design Works. The two signals address the two failure modes of static reflection (Figure 1): $\tilde{H}^{(t)}$ replaces *blind reflection* with a trigger that fires only while the pool still disagrees, and $\Delta^{(t)}$ replaces the *uniform schedule* by withholding termination until an apparent consensus has stopped moving (Eq. 8).

Two-Signal Stability Correlates with Reasoning Quality. We further verify that the controller’s two-signal classification tracks reasoning quality. On MATH-500 with Qwen3-32B, we label each query at termination as *stable* ($\tilde{H}^{(t^*)} < \tau_H$ and $\Delta^{(t^*)} < \tau_\Delta$) or *other*. As shown in Figure 3, the two states correlate with correctness in opposite directions: **81.2%** of correct answers terminate as stable, whereas **69.7%** of incorrect answers terminate as other. We stress that this is a correlation and not a guarantee: read in the other direction, **30.3%** of incorrect answers still terminate in the stable state (false confidence) and **18.8%** of correct answers are judged unstable. The joint condition should therefore be understood as a *correlational runtime indicator of reasoning reliability*—useful as evidence that the controller’s stability classification tracks *cognitive cohesion*, but far from a certificate of correctness. Under the intrinsic, oracle-free setting studied here, no such certificate is attainable, since both signals measure disagreement and inter-round movement rather than correctness itself.

Adaptability Across Modern LLMs. Table 4 reveals complementary benefits across model scales. On weaker models such as Qwen3-8B, multi-perspective initialization contributes the most, compensating for the limited intrinsic diversity of single-pass sampling. On stronger models such as GPT-4o and DeepSeek-V3, Belief Shift contributes the most: these models articulate apparent consensus more convincingly, making unstable low-entropy states more prevalent and the stability check more critical. REGAR thus functions as a model-agnostic reflection-control layer that adapts its dominant contribution to the underlying LLM.

Computational Efficiency via Adaptive Resource Allocation. Unlike fixed-budget alternatives such as Self-Consistency that sample k paths regardless of difficulty, REGAR adopts an *adaptive computation* strategy: easy problems entering the stable state terminate early, while fragmented or unstable instances invoke reflection and consume more rounds. This yields the 26–30% token reductions over the strongest reflective baselines (Tables 1, 2) while still delivering the highest accuracy across all five models. Since fewer tokens do not automatically imply lower latency, Appendix G further reports per-problem LLM call counts and the subset of calls lying on the serial critical path, where REGAR is also the cheapest among all reflective methods.

6 Conclusion

We investigate why LLM self-correction often fails without external feedback and find that toxic reflection occurs as frequently as effective correction. We attribute this to the shortsighted nature of static reflection pipelines and propose two complementary signals to address it: Reflecting Entropy, which captures snapshot disagreement among reasoning trajectories, and Belief Shift, which captures inter-round belief revision. The two signals drive a two-stage hierarchical controller, instantiated as the REGAR framework with a single, uniform reflection primitive as its corrective action. Experiments on mathematical and commonsense reasoning across five contemporary LLMs show consistent improvements over strong baselines while reducing unnecessary reflection, validating the two-signal hierarchical design as an effective control mechanism for LLM self-correction.

Limitations

While the REGAR framework demonstrates consistent improvements, several limitations remain. Our approach operates entirely through prompt engineering without modifying model parameters, meaning the uncertainty awareness is orchestrated externally rather than internalized by the model itself. A more principled direction would involve training models to natively perceive Reflecting Entropy and Belief Shift—for instance, by incorporating signal-based rewards in RLHF or using them as supervision during instruction tuning—enabling autonomous regulation of reflection intensity.

A more fundamental limitation is that REGAR operates in the *intrinsic*, oracle-free regime, where the arbiter is the same model that generates the trajectories. Our two signals measure *disagreement* and *inter-round movement*, not correctness, so when the base model is systematically wrong across all sampled perspectives on a class of problems, the pool can converge to a stable, low-entropy consensus on the wrong answer, and neither signal can detect it. Three design choices partially mitigate—but do not eliminate—this circularity: the stop/continue decision is made from pool-level statistics across multiple trajectories rather than from the arbiter’s self-judgement, so a systematic error contaminating only part of the pool leaves residual disagreement that keeps the controller reflecting; multi-perspective initialization seeds the pool with structurally distinct solution routes rather than i.i.d. temperature samples, reducing the chance that a single bias dominates from the outset; and the arbiter is not confined to endorsing an existing cluster, since it may emit a third answer disagreeing with both leading clusters (Appendix D), while the equivalence checking that feeds the signals is a deterministic symbolic/rule-based cascade (Appendix B) that falls back to a separate small model only when needed. Nonetheless, this bounds the applicability of any oracle-free reflection controller, REGAR included: intrinsic self-correction cannot recover a correct answer that the base model never produces.

Finally, our experiments focus on tasks with well-defined correctness criteria; the applicability to open-ended domains where multiple valid answers exist remains unexplored.

Acknowledgments

We thank the anonymous reviewers and the area chair for their valuable feedback, which substantially improved this paper.

References

- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. [Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks](#). *Preprint*, arXiv:2211.12588.
- Xuhang Chen, Zhifan Song, Deyi Ji, Shuo Gao, and Lanyun Zhu. 2026. [Self-signals driven Multi-LLM debate for efficient and accurate reasoning](#). *Preprint*, arXiv:2510.06843.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-First International Conference on Machine Learning*.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujia Yang, Nan Duan, and Weizhu Chen. 2023. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.
- Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Liping Liu, Chunhong Zhang, Likang Wu, Chuang Zhao, Zheng Hu, Ming He, and Jianping Fan. 2025. Instruct-of-reflection: Enhancing large language models iterative reflection capabilities via dynamic-meta instruction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9956–9978, Albuquerque, New Mexico. Association for Computational Linguistics.
- Guanran Luo, Wentao Qiu, Zhongquan Jian, Meihong Wang, and Qingqiang Wu. 2025. [Uncertainty quantification via reasoning–explanation symmetry in LLMs](#). OpenReview preprint.
- Ruotian Ma, Peisong Wang, Cheng Liu, Xingyan Liu, Jiaqi Chen, Bang Zhang, Xin Zhou, Nan Du, and Jia Li. 2025. [SS²R: Teaching LLMs to self-verify and self-correct via reinforcement learning](#). *Preprint*, arXiv:2502.12853.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.
- Mathematical Association of America. 2024. AIME: American Invitational Mathematics Examination. Public competition problem set. We use a curated subset of historical AIME problems for evaluation.
- Kumpati S Narendra and Mandayam AL Thathachar. 2012. *Learning automata: an introduction*. Courier corporation.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [GPT-4 Technical Report](#). *Preprint*, arXiv:2303.08774.
- Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1100–1126, St. Julian’s, Malta. Association for Computational Linguistics.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [LLaMA: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Ken Tsui. 2025. [Self-Correction Bench: Uncovers and Addressing the Self-Correction Blind Spot in Large Language Models](#). *Preprint*, arXiv:2507.02778.
- Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908, Bangkok, Thailand. Association for Computational Linguistics.
- Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint*.
- Keyu Wang, Jin Li, Shu Yang, Zhuoran Zhang, and Di Wang. 2025. [When truth is overridden: Uncovers the internal origins of sycophancy in large language models](#). *Preprint*, arXiv:2508.02087.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances*

in *Neural Information Processing Systems*, 35:24824–24837.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. 2024. [Simple synthetic data reduces sycophancy in large language models](#). *Preprint*, arXiv:2308.03958.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Zongqian Wu, Baoduo Xu, Ruochen Cui, Mengmeng Zhan, Xiaofeng Zhu, and Lei Feng. 2025. [Rethinking chain-of-thought from the perspective of self-training](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. *ArXiv*.

Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. 2025. Understanding the dark side of LLMs’ intrinsic self-correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27066–27101, Vienna, Austria. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [OPT: Open pre-trained transformer language models](#). *Preprint*, arXiv:2205.01068.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through inconsistent solving perspectives. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3602–3622, Bangkok, Thailand. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

A Notation

Table 5 collects every symbol used in §3.1–§3.3, together with its meaning and range. Symbols are grouped by the component they belong to: the trajectory pool, the two uncertainty signals, and the controller and its update rule.

Symbol	Meaning	Range / Value
<i>Trajectory pool (§3.1)</i>		
x	Input query	—
t	Reflection round index	$\{0, \dots, T_{\max}\}$
$TP^{(t)}$	Trajectory pool at round t	set of $(y_i, w_i^{(t)})$
y_i	i -th reasoning trajectory	$(s_i^1, \dots, s_i^{L_i})$
s_i^j	j -th reasoning step of y_i	—
L_i	Number of steps in y_i	≥ 1
$w_i^{(t)}$	Heuristic confidence weight	$[0, 1], \sum_i w_i^{(t)} = 1$
N	Current pool size	grows with t
$\mathcal{I}, I_k$	Perspective instructions	$ \mathcal{I} = m$
m	Number of perspectives	2–6 (model-decided)
<i>Uncertainty signals (§3.2)</i>		
d	Normalized prefix depth	$[0, 1]$
$\mathcal{D}$	Depth set used	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$
$y_i^{[d]}$	Depth- d prefix of y_i	—
C_k^d	k -th cluster at depth d	—
K_d	Number of clusters at depth d	typically ≤ 4
$P^{(t)}(C_k^d)$	Cluster belief mass (Eq. 2)	$[0, 1]$
$H_d^{(t)}$	Raw depth- d entropy	$[0, \log K_d]$
$\tilde{H}_d^{(t)}$	Normalized depth- d entropy	$[0, 1]$
$\tilde{H}^{(t)}$	Reflecting Entropy (Eq. 5)	$[0, 4.36]$
γ	Depth discount	$(0, 1]; 0.7$
$\mathcal{P}^{(t)}$	Endpoint cluster distribution	simplex
$\Delta^{(t)}$	Belief Shift (Eq. 7)	$[0, \log 2]$
<i>Controller and updates (§3.3)</i>		
τ_H	Reflecting Entropy threshold	0.55
τ_Δ	Belief Shift threshold	0.10
$T_{\max}$	Iteration budget	4
C_A, C_B	Top-two endpoint clusters	—
y_A, y_B	Their highest-weight members	—
y_{new}	Arbiter’s refined trajectory	—
$\mathcal{S}^+ / \mathcal{S}^-$	Positive / negative feedback tag	—
α	LRP reinforcement rate	$(0, 1); 0.25$
ρ	LRP decay factor	$(0, 1); 0.5$
ϵ	Pruning / padding threshold	10^{-6}

Table 5: Notation used throughout §3.1–§3.3. Note that $\tilde{H}^{(t)}$ is a discount-weighted sum over $\mathcal{D}$ and is *not* renormalized across depths, so its upper bound is $\sum_{d \in \mathcal{D}} \gamma^{1-d} \approx 4.36$ at $\gamma=0.7$ rather than 1.

B Trajectory Clustering: Implementation Details

This appendix provides the full procedural details of the depth- d trajectory clustering used by both Reflecting Entropy and Belief Shift in §3.2. As stated in the main text, equivalence is determined at the trajectory-conclusion level rather than over token embeddings, through a three-level cascade ordered from deterministic precision to broad coverage.

Level 1 (Symbolic Verification). For tasks whose answers admit a symbolic representation (mathematical, scientific), we extract the final-answer expression from each prefix via task-specific patterns and decide equivalence by `sympy.simplify( $a_i - a_j$ ) = 0`, with a numerical fallback for non-simplifiable cases.

Level 2 (Rule-based Normalization and State Match). For structured outputs (boolean, multiple-choice, normalized strings) equivalence is decided by canonicalization plus string equality. For intermediate-depth prefixes ($d < 1.0$) of Program-of-Thought trajectories, the prefix is executed in a sandboxed environment and the resulting variable-state multiset is compared elementwise.

Level 3 (Lightweight Equivalence Judge). Only when L1 and L2 are inapplicable do we invoke a small auxiliary model (Qwen2.5-1.5B-Instruct, deployed locally) restricted to a binary yes/no equivalence prompt. This judge does not reason about the task; it only checks whether two conclusions are semantically equivalent, and is deliberately kept independent of the generator/arbiter so that the LLM responsible for reasoning never doubles as the equivalence checker.

Depth-Dependent Strictness. The strictness of equivalence varies with depth: at $d = 1.0$ the loosest check (symbolic value) is used so that semantically equivalent answers reached by different routes are merged, while at shallow depths stricter rule-based or state-multiset checks preserve genuinely divergent reasoning routes.

Online Clustering Procedure. The cascade is integrated into a canopy-style online clustering procedure (Algorithm 1) with $O(N \cdot K)$ judgment calls per depth, where K is the number of clusters (typically $K \leq 4$ in our experiments).

Cross-Round Cluster Alignment. For Belief Shift (Eq. 7), cluster identifiers are arbitrary across rounds, so computing $\Delta^{(t)}$ requires aligning each round- t cluster against round- $(t-1)$ clusters via the same equivalence cascade above, with newly

Algorithm 1 Hierarchical Trajectory Clustering at Depth d

Require: Trajectory prefixes $\{y_i^{[d]}\}_{i=1}^N$, depth d
Ensure: Cluster assignment $\{c_i\}_{i=1}^N$
1: $C \leftarrow \emptyset$ ▷ cluster representatives
2: **for** $i = 1$ to N **do**
3: $\text{matched} \leftarrow \text{False}$
4: **for** each $r \in C$ **do**
5: **if** $\text{EQUIV}_{L1}(y_i^{[d]}, r) = \text{True}$ **then**
6: $c_i \leftarrow r$; $\text{matched} \leftarrow \text{True}$; **break**
7: **else if** $\text{EQUIV}_{L2}(y_i^{[d]}, r) = \text{True}$ **then**
8: $c_i \leftarrow r$; $\text{matched} \leftarrow \text{True}$; **break**
9: **else if** $\text{EQUIV}_{L3}(y_i^{[d]}, r) = \text{True}$ **then**
10: $c_i \leftarrow r$; $\text{matched} \leftarrow \text{True}$; **break**
11: **end if**
12: **end for**
13: **if not matched then**
14: $C \leftarrow C \cup \{y_i^{[d]}\}$; $c_i \leftarrow y_i^{[d]}$
15: **end if**
16: **end for**
17: **return** $\{c_i\}$

emerging clusters padded by a small mass $\epsilon = 10^{-6}$ on the missing side to keep the JSD finite.

C Experimental Details

C.1 Decoding and Scoring Protocol

Because Qwen3 is a hybrid reasoning model, we explicitly *disable* its thinking mode (`enable_thinking=False` / `/no_think`) for all Qwen3 models and for every method, including all baselines and REGAR. Thinking mode is itself a form of long internal chain-of-thought self-reflection, and leaving it enabled would confound the model’s internal reflection with the externally scheduled reflection mechanism under study; disabling it also places the five heterogeneous model families on a common footing.

Beyond this, we adopt a deliberately conservative and uniform protocol for every method: zero-shot prompting with no few-shot exemplars, a single low-temperature ($T=0.3$) sample rather than multi-sample decoding, and strict automatic answer matching (numeric rounding and canonical string equality) without any model-based grader. This protocol yields lower absolute accuracies than the more permissive setups typically used in public reports, which commonly rely on few-shot prompting, self-consistency, or model-based graders. Since all baselines and REGAR are evaluated under the identical protocol with the identical scorer, however, the relative comparisons we draw—and hence every claim in §4.4 and §4.5—are unaffected.

C.2 Datasets

We conduct experiments on four representative benchmarks spanning mathematical and commonsense reasoning tasks.

Mathematical Reasoning. We select three benchmarks of increasing difficulty. **MATH-500** (Lightman et al., 2024) is a competition-level subset covering algebra, geometry, number theory and pre-calculus. **AIME** (Mathematical Association of America, 2024) consists of olympiad-level problems from the American Invitational Mathematics Examination, requiring multi-step deductive reasoning. **GSM8K** (Cobbe et al., 2021) comprises grade-school math word problems with multi-step arithmetic chains, retained as a saturated easy-task control.

Commonsense Reasoning. We employ **StrategyQA** (Geva et al., 2021), a question-answering benchmark that requires implicit multi-hop reasoning over commonsense knowledge with binary yes/no answers, evaluating reasoning coherence beyond mathematical domains.

Table 6 summarizes the statistics of these benchmarks.

Dataset	Size	Task Category	Ans. Format
MATH-500	500	Math Reasoning	Numeric
AIME	933	Math Reasoning	Numeric
GSM8K	1,319	Math Reasoning	Numeric
StrategyQA	2,290	Commonsense QA	Boolean

Table 6: Statistics of the four reasoning benchmarks.

C.3 Hyperparameter Settings

Table 7 summarizes the key hyperparameters used in our experiments.

Hyperparameter	Symbol	Value
Number of perspectives	m	2–6
Temperature	T	0.3
Depth set	$\mathcal{D}$	$\{0.2, 0.4, 0.6, 0.8, 1.0\}$
Depth discount	γ	0.7
Reflecting Entropy threshold	τ_H	0.55
Belief Shift threshold	τ_Δ	0.10
Reinforcement rate	α	0.25
Decay factor	ρ	0.5
Max iterations	$T_{\max}$	4
L3 equivalence judge	—	Qwen2.5-1.5B-Instruct

Table 7: Default hyperparameter settings of REGAR used throughout the experiments.

We elaborate on the selection of critical hyperparameters as follows. During the initialization phase,

we constrain the LLM to generate between 2 and 6 distinct reasoning perspectives for each query, balancing exploration of the initial solution space against computational overhead. The depth set $\mathcal{D} = \{0.2, 0.4, 0.6, 0.8, 1.0\}$ partitions each trajectory into five evenly spaced prefixes, and the depth discount $\gamma = 0.7$ in Eq. 5 grants the largest weight to the endpoint depth while still admitting non-trivial signal from earlier depths. The two thresholds τ_H and τ_Δ jointly determine the controller’s three states: $\tau_H = 0.55$ separates fragmented from converged snapshots, while $\tau_\Delta = 0.10$ separates stable from unstable belief distributions; both values were selected via grid search on a held-out development split of AIME that is disjoint from the evaluation sets used in Table 1, then frozen and reused unchanged across all five models and four benchmarks, where they yield consistent performance across datasets and model scales. Upon receiving feedback signals from each reflection round, we apply the Linear Reward–Penalty (LRP) mechanism to update trajectory weights: the reinforcement rate α and decay factor ρ govern how strongly positive and negative feedback affect the weights, respectively, with $\alpha = 0.25$ and $\rho = 0.5$ providing the most stable and effective dynamics in our experiments.

C.3.1 Sensitivity Analysis

We probe the robustness of three hyperparameters whose settings reflect non-trivial design trade-offs: the number of perspectives m , the depth discount γ , and the threshold pair (τ_H, τ_Δ) . All sensitivity experiments are conducted on a *held-out development split* of AIME (the most discriminative benchmark in our setup) that is disjoint from the test set used in Table 1, with Qwen3-32B and every other hyperparameter held at the default value reported in Table 7. We emphasize that no threshold was ever selected on an evaluation set; the single operating point reported above is then applied unchanged to all five models and four benchmarks.

Number of Perspectives m . Figure 4 sweeps m from 2 to 8. Accuracy rises sharply on both AIME and MATH-500 up to $m=4$ and enters a plateau between $m=4$ and $m=6$, while #Tokens grows almost linearly with m . Beyond $m=6$ accuracy begins to recede slightly, indicating that additional perspectives saturate the achievable diversity at this query difficulty and only inflate cost. The default dynamic range $m \in [2, 6]$ thus captures more than 99% of the achievable accuracy at less than 70% of

the cost incurred at $m=8$.

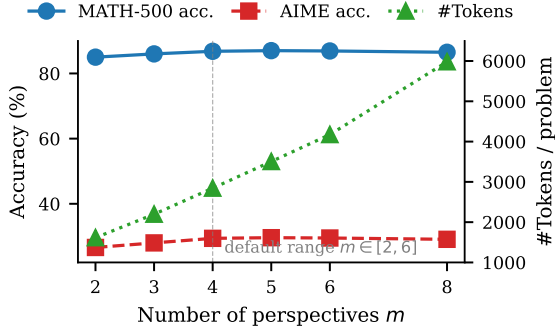


Figure 4: Sensitivity to the number of perspectives m on Qwen3-32B (AIME and MATH-500 accuracy, left axis; #Tokens, right axis). Vertical line marks the default range $m \in [2, 6]$.

Depth Discount γ . Figure 5 sweeps γ over $\{0.3, 0.5, 0.7, 0.9, 1.0\}$. The accuracy curve exhibits an inverted-U shape peaking at $\gamma \in [0.6, 0.8]$: small γ over-amplifies shallow-depth noise, while $\gamma=1.0$ collapses the discount-weighted aggregation to uniform averaging and erases the endpoint emphasis. The horizontal dashed line marks the “ $d=1.0$ only” baseline (i.e., disabling multi-depth aggregation), which sits below every $\gamma < 1.0$ point—confirming that earlier depths carry non-trivial signal that the default $\gamma=0.7$ correctly weights.

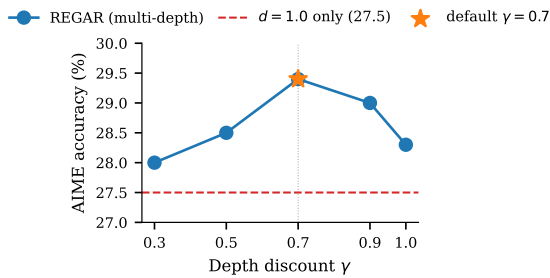


Figure 5: Sensitivity to the depth discount γ on AIME (Qwen3-32B). The dashed line marks the “ $d=1.0$ only” reference (no multi-depth aggregation); the star marks the default $\gamma=0.7$.

Joint Threshold Sensitivity (τ_H, τ_Δ). Because the two thresholds gate adjacent stages of the same controller (Eq. 8), we sweep them jointly on a 5×5 grid. Figure 6 reveals a broad accuracy plateau centered at the default $(0.55, 0.10)$ that tolerates ± 0.10 in τ_H and ± 0.05 in τ_Δ with at most a 0.3% accuracy fluctuation. Performance degrades sharply only when $\tau_H \geq 0.75$ (the controller becomes too lenient and fragmented snapshots evade

reflection) or $\tau_\Delta \geq 0.25$ (the stability check ignores genuine pseudo-consensus and terminates prematurely), consistent with the qualitative roles assigned to the two signals in §3.3.

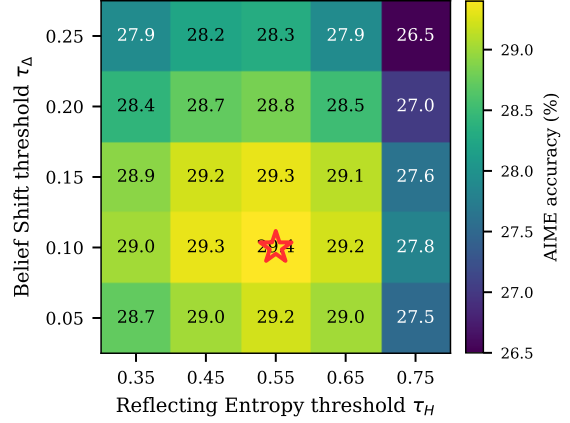


Figure 6: AIME accuracy (Qwen3-32B) over a 5×5 grid of (τ_H, τ_Δ) . The red star marks the default operating point $(0.55, 0.10)$.

Cross-Setting Stability of the Thresholds. The sweeps above are run on a single model–task combination (AIME / Qwen3-32B), so they establish that the operating point sits on a plateau *in that setting* rather than demonstrating cross-setting robustness directly. The strongest evidence we can offer for the latter is the main experimental protocol itself: the *same* frozen operating point ($\tau_H=0.55$, $\tau_\Delta=0.10$, $\gamma=0.7$), selected once on the AIME development split, is reused unchanged across all five models and four benchmarks in Tables 1, 2, and 8, and REGAR remains the best method in every one of those 5×4 configurations. Had the thresholds been over-fit to AIME / Qwen3-32B, a single globally fixed point could not have held across settings that differ this widely in model scale, model family, and task type. We nonetheless note that this is indirect evidence, and that a per-setting sweep might reveal a better task-specific operating point than the single global one we deliberately committed to.

Other hyperparameters. For the LRP rates (α, ρ) , the temperature T , and the iteration budget $T_{\max}$, we additionally verified that $\pm 50\%$ perturbations around the defaults change AIME accuracy by less than 0.3%, indicating that these settings are not the load-bearing components of the framework.

D Prompt Templates

D.1 Prompt for Perspective Generation

Different reasoning tasks may benefit from diverse problem-solving perspectives. We design a self-curated prompt generation process, enabling LLMs to dynamically generate task-specific reasoning perspectives based on the input query. The prompts for perspective generation are as follows:

Mathematical Reasoning Task:

You are a math specialist who specializes in mathematical problem-solving from diverse perspectives. Given a math question, you need to carefully analyze the question and dynamically generate several useful prompt instructions. These prompt instructions should be diverse and also useful for math-solving. These prompt instructions are used to guide the language model to think in different ways, pay attention to different emphases, and reason from different perspectives for more accurate math solving.

For instance, you can adopt multi-faceted thinking (logical thinking, lateral thinking, analogical thinking, etc.), different reasoning perspectives (e.g., top-down, bottom-up, step-by-step), and different emphases of concern (entity words, numbers, units, percentages, math knowledge, etc.) for input question in prompt instruction.

Here are some guidance rules for Prompt Generation:

1. Tone Requirement: Please generate prompt instructions in the third person.
2. Content Requirement: Each prompt instruction should adopt a different way of thinking, or focus on a different perspective, or different emphases to solve the question.
3. Number Requirement: Dynamically generate the most valuable 2 to 6 prompt instructions based on the input math question.
4. Format Requirement: Each prompt instruction should start with ### and end with @@@.

The input math question is {Question}. Please generate the most suitable prompts:

Commonsense Reasoning Task:

You are a reasoning specialist who excels at commonsense question-answering from diverse perspectives. Given a commonsense question, you need to carefully analyze the question and dynamically generate several useful prompt instructions. These prompt instructions should be diverse and effective for commonsense reasoning. These prompt instructions are used to guide the language model to think in different ways, leverage different knowledge sources, and reason from different perspectives for more accurate answering.

For instance, you can adopt multi-faceted thinking (causal reasoning, temporal reasoning, spatial reasoning, etc.), different reasoning strategies (e.g., decomposition, analogy, elimination), and different knowledge emphases (world knowledge, social norms, physical

laws, etc.) for the input question in prompt instruction.

Here are some guidance rules for Prompt Generation:

1. Tone Requirement: Please generate prompt instructions in the third person.
2. Content Requirement: Each prompt instruction should adopt a different way of thinking, or focus on a different knowledge domain, or different reasoning strategy to answer the question.
3. Number Requirement: Dynamically generate the most valuable 2 to 6 prompt instructions based on the input question.
4. Format Requirement: Each prompt instruction should start with ### and end with @@@.

The input commonsense question is {Question}. Please generate the most suitable prompts:

D.2 Prompt for Reflection

When the controller invokes reflection, the system passes the top-two cluster representatives y_A, y_B to the LLM arbiter (the same model used to generate trajectories) through the following single prompt. The arbiter analyzes errors in each path and produces a refined trajectory; we deliberately omit any reference to the current belief distribution so that the arbiter's verdict is not biased by which side currently dominates the pool. Task-specific prompts are listed below.

Reflection (Mathematical Reasoning):

You are an impartial mathematical arbiter. Two candidate solution trajectories are provided for the same question. Your task is to (i) analyze the reasoning errors in each trajectory and (ii) produce a single refined solution as your conclusion. The refined solution may agree with Trajectory A, agree with Trajectory B, or constitute a new third answer if both trajectories contain significant flaws.

Original Question: {Question}

Trajectory A: {Trajectory_A}

Trajectory B: {Trajectory_B}

Please evaluate both trajectories based on:

1. Computational Correctness: Are the calculations in each trajectory accurate?
2. Logical Rigor: Is each reasoning chain mathematically sound?
3. Methodological Appropriateness: Is the chosen approach suitable for this problem?

Output Format: Provide your response in three parts:

Errors in Trajectory A: List the specific reasoning or computational errors in A (write "None" if A is sound).

Errors in Trajectory B: List the specific reasoning or computational errors in B (write "None" if B is sound).

Refined Solution: Provide a complete reasoning trajectory leading to the final answer. Indicate whether this solution agrees with A, agrees with B, or is a new third answer.

Reflection (Commonsense Reasoning):

You are an impartial commonsense reasoning arbiter. Two candidate solution trajectories are provided for the same question. Your task is to (i) analyze the reasoning errors in each trajectory and (ii) produce a single refined solution as your conclusion. The refined solution may agree with Trajectory A, agree with Trajectory B, or constitute a new third answer if both trajectories contain significant flaws.

Original Question: {Question}

Trajectory A: {Trajectory_A}

Trajectory B: {Trajectory_B}

Please evaluate both trajectories based on:

1. Factual Grounding: Does each trajectory rely on accurate real-world knowledge?
2. Logical Soundness: Is each reasoning chain coherent and well-structured?
3. Commonsense Alignment: Does each conclusion align with everyday understanding?

Output Format: Provide your response in three parts:

Errors in Trajectory A: List the specific reasoning or factual errors in A (write “None” if A is sound).

Errors in Trajectory B: List the specific reasoning or factual errors in B (write “None” if B is sound).

Refined Solution: Provide a complete reasoning trajectory leading to the final answer. Indicate whether this solution agrees with A, agrees with B, or is a new third answer.

E Results on GSM8K (Saturated Easy-Task Control)

Methods	Q3-8B	Q3-32B	L3-70B	DS-V3	GPT-4o	#Tokens
CoT	88.5	93.2	94.5	93.8	94.6	507
PoT	89.2	94.0	94.8	94.2	94.9	474
CRITIC	88.8	93.5	95.1	94.5	95.0	3182
Self-Consistency ($k=8$)	90.5	94.7	95.6	95.0	95.5	4086
SC (budget-matched)	89.5	94.0	95.2	94.5	95.1	2913
Multi-Agent Debate	89.0	93.8	95.0	94.6	95.0	3937
Self-Contrast	89.8	94.3	95.4	94.8	95.3	3486
IoRT	90.6	94.9	95.8	95.2	95.7	3801
IoRT (budget-matched)	89.7	94.2	95.3	94.6	95.2	2904
REGAR (Ours)	91.0	95.2	96.0	95.5	95.9	2829

Table 8: Accuracy (%) and #Tokens on GSM8K across five LLMs. **Bold:** best accuracy per column; in the #Tokens column, the lowest among *reflective* methods.

To verify that REGAR does not regress on simple problems where reflection has limited room to help, we additionally evaluate all methods on GSM8K under the same setup as the main experiments (§C.3). Results are reported in Table 8. Across all five models, REGAR retains a small but consistent improvement (0.2–0.4%) over the strongest baseline, while consuming substantially fewer total tokens than Self-Consistency and IoRT. The narrow accuracy band on GSM8K—where even the non-reflective CoT baseline already exceeds 88%—is consistent with the dataset being

saturated for modern LLMs; the absence of degradation confirms that the hierarchical controller correctly terminates early on easy instances rather than over-applying reflection.

F Uncertainty-Guided Baselines

§2.2 distinguishes REGAR from prior uncertainty-based work on the grounds that existing signals are single-snapshot and endpoint-only. As neither method reports results under our protocol, we re-implement both under the exact Table 1 setting (all five LLMs, identical decoding and scorer). Our **SE-CoT** implementation follows Wu et al. (2025): it selects a task-specific initial prompt by predictive uncertainty, then re-samples and halts once the sampled answers reach a low-entropy consensus. Our **RES** implementation follows Luo et al. (2025), scoring the reliability of a single answer by the agreement between its forward reasoning and a backward, answer-conditioned explanation, without iterative reflection.

Both trail REGAR on every model column of Table 1: on Qwen3-8B, SE-CoT scores 61.9/12.7 and RES 60.9/12.0 on MATH-500/AIME versus REGAR’s 65.2/17.5, and the gap persists on the four larger models (e.g., DS-V3: SE-CoT 81.6/20.4, RES 80.6/19.5, REGAR 84.7/26.2). SE-CoT lands just below the $k=8$ Self-Consistency baseline: in our implementation its low-entropy gate rarely fires a second iteration (≈ 1.05 rounds on average), leaving it close to a single-pass method in practice.

This shortfall is consistent with the structural gap our two signals target: both baselines score only the final answer within a single round, so neither can express (i) divergence that arises early in reasoning and later collapses into surface agreement, nor (ii) whether an already-converged pool is still being rewritten across rounds. REGAR covers both by construction—multi-depth Reflecting Entropy is evaluated at prefix depths below the endpoint (Table 13), and Belief Shift compares consecutive rounds explicitly. SID (Chen et al., 2026), the third method cited in §2.2, is likewise a snapshot-only signal, designed for multi-agent debate; we leave a faithful re-implementation to future work.

G Latency and Call-Count Analysis

Total token consumption (Tables 1, 2) measures computational cost but not necessarily wall-clock time, since a method issuing few tokens across many *sequential* round-trips can still be slow. Be-

fore turning to latency, we state the token accounting precisely: #Tokens Avg. covers *every* stage of the pipeline—perspective generation, the m initial trajectories, each reflection round’s arbiter prompt *and* output, and the equivalence checks—and counts both input and output tokens, averaged over five independent runs, with nothing excluded. The same accounting is applied to every baseline.

We additionally report, in Table 9, the average number of LLM calls per problem (#Calls), the same efficiency proxy used in prior reflective-reasoning work, together with the number of calls that lie on the *serial critical path*—the true latency determinant, since parallelizable draws (Self-Consistency’s k samples, REGAR’s m initial trajectories) collapse into a single wall-clock step.

Method	#Calls Avg.	#Serial (latency)	#Tokens Avg.
CoT	1	1	493
PoT	1	1	471
Self-Consistency ($k=8$)	8	1 (parallel)	4063
Self-Contrast	7.8	7.8	3554
Multi-Agent Debate	9	3	3946
CRITIC	9	9	3127
IoRT	7.3	7.3	3827
REGAR (Ours)	6.5	4.2	2843

Table 9: Per-problem LLM call counts and token consumption on MATH-500. #Calls Avg. is the total number of LLM calls; #Serial counts only calls on the serial critical path, since parallelizable draws collapse to one wall-clock step. Baseline #Calls follow the accounting reported in the respective prior work. All values are averages over five runs; REGAR’s are non-integer because its adaptive controller issues a problem-dependent number of calls.

REGAR’s call accounting follows the same convention as the baselines: one perspective-generation call, m initial trajectory calls, and R reflection rounds, with $m \approx 3.3$ (typically 3–5) and $R \approx 2.2$ on average under the adaptive controller (bounded by $T_{\max}=4$). The lightweight L3 equivalence judge runs on a separate small model (Appendix B) and is excluded, consistent with auxiliary verifiers likewise being excluded in the baseline accounting.

Three observations follow. (i) **Call count.** Among all reflective and multi-call methods, REGAR issues the fewest LLM calls per problem (6.5, versus IoRT 7.3, Self-Contrast 7.8, Self-Consistency 8, and Multi-Agent Debate and CRITIC at 9), coherent with it also being the most

token-efficient. (ii) **Latency.** On the serial critical path REGAR needs only 4.2 calls, fewer than every serial reflective baseline, because its m initial trajectories are generated in parallel and only the $R \approx 2.2$ reflection rounds are inherently sequential. Only non-reflective CoT/PoT and embarrassingly-parallel Self-Consistency have lower latency, and they pay for it in accuracy (CoT/PoT) or in $\sim 1.4\times$ the tokens (4063 vs. 2843) at lower accuracy (SC). As a concrete instantiation on Qwen3-32B with single-node vLLM serving, this translates to roughly **29 s** per problem for REGAR versus **48 s** for IoRT. (iii) **Adaptivity.** REGAR’s calls are bounded by $T_{\max}=4$ and are adaptively fewer on easier instances—on the saturated GSM8K many problems terminate at round 0–1 (cf. Table 10)—so the reflection overhead is paid only where reflection is actually needed. Fewer tokens therefore do translate into lower latency in our setting.

H Case Study

We illustrate the two-stage hierarchical controller (§3.3) with four representative cases (GPT-4o, $T_{\max}=4$, $\tau_H=0.55$, $\tau_\Delta=0.10$), one per benchmark. Each table records the question, the initial pool $TP^{(0)}$ of $m=3$ trajectories generated from *three distinct perspectives* via the prompt template in Appendix D, per-round controller signals (Reflecting Entropy $\tilde{H}^{(t)}$ and Belief Shift $\Delta^{(t)}$), the LLM arbiter’s reflection on the top-two endpoint clusters C_A, C_B , and the weight update via the Linear Reward–Penalty rule (Eq. 10). Together they illustrate four representative controller trajectories: (i) immediate TERMINATE when the pool is already a fully converged consensus at every depth (Table 10); (ii) REFLECT via Stage 1 followed by a rapid collapse to consensus (Table 11); (iii) REFLECT via Stage 1 sustained over multiple rounds on a hard problem with three competing answers (Table 12); and (iv) REFLECT triggered *solely* by shallow-depth divergence while the final answers are unanimous (Table 13), the case an endpoint-only signal cannot detect. All reported signal values follow Eqs. 2–7 with $\gamma=0.7$ and $\mathcal{D} = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, so that $\tilde{H}^{(t)}$ ranges over $[0, \sum_{d \in \mathcal{D}} \gamma^{1-d}] = [0, 4.36]$ and each value can be recomputed from the cluster structure stated alongside it. Trajectory excerpts are slightly condensed from raw model outputs, preserving the actual perspectives, intermediate computations, and arbiter verdicts.

Iterative Reflection Process	Output
<p>Question: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?</p> <p>Generated Perspectives ($m=3$):</p> <p>I_1 [top-down]: “Decompose the total into blue and white components, then sum.”</p> <p>I_2 [bottom-up]: “Compute each fiber type from the given numerical ratio, then aggregate.”</p> <p>I_3 [unit-focused]: “Track the unit ‘bolt’ and the modifier ‘half that much’ explicitly to avoid scale errors.”</p> <p>Initial Pool $TP^{(0)}$ (PoT executions):</p> <ul style="list-style-type: none"> • y_1 (I_1): blue = 2; white = blue/2; answer = blue + white → 3.0 • y_2 (I_2): white_ratio = 0.5; white = 2*white_ratio; total = 2 + white → 3.0 • y_3 (I_3): blue_bolts = 2; # half of 2 bolts; white_bolts = 1; total_bolts = 3 → 3.0 <p>Final Answers: [3.0, 3.0, 3.0] Initial Weights: $w_i^{(0)} = 1/3$ for all i.</p> <p>Controller Signals ($t=0$): the three trajectories collapse into a single endpoint cluster, and at every depth $d \in \mathcal{D}$ they also share the same intermediate steps, so $\tilde{H}_d^{(0)} = 0$ for all d and $\tilde{H}^{(0)} = 0.00 < \tau_H$. By convention $\Delta^{(0)} \triangleq 0$.</p> <p>Decision (Eq. 8, Stage 2): $\tilde{H}^{(0)} < \tau_H$ and $\Delta^{(0)} < \tau_\Delta \Rightarrow$ TERMINATE. The arbiter is never invoked; the highest-weight trajectory in the dominant endpoint cluster is emitted.</p>	<p>Round 0</p> <p>3.0 (✓)</p>

Table 10: A GSM8K case (GPT-4o). Three independent perspectives (top-down, bottom-up, unit-focused) yield identical reasoning traces and the same final answer, driving $\tilde{H}^{(0)}$ to zero. The two-stage rule terminates immediately at round $t=0$ via Stage 2, illustrating REGAR’s cost-free behaviour on already-converged easy instances; this is consistent with the saturated GSM8K results reported in Appendix E.

Iterative Reflection Process	Output
<p>Question: If $x = 2$ and $y = 5$, what is the value of $\frac{x^4 + 2y^2}{6}$?</p> <p>Generated Perspectives ($m=3$):</p> <p>I_1 [direct substitution]: “Substitute the given values into the expression and evaluate arithmetically.”</p> <p>I_2 [stepwise simplification]: “Compute x^4 and $2y^2$ separately, sum them, then divide by 6.”</p> <p>I_3 [symbolic manipulation]: “Treat the expression as a polynomial form and use a symbolic library to ensure exactness.”</p> <p>Initial Pool $\text{TP}^{(0)}$ (PoT executions):</p> <ul style="list-style-type: none"> • y_1 (I_1): $x=2$; $y=5$; $\text{answer} = (x**4 + 2*y**2)/6 \rightarrow \mathbf{11.0}$ • y_2 (I_2): $a = 2**4 \# 16$; $b = 2*5**2 \# 50$; $\text{answer} = (a + b)/6 \rightarrow \mathbf{11.0}$ • y_3 (I_3): $\text{from sympy import Rational; answer = Rational}(2**4 + 2*5**2, 6) \rightarrow \mathbf{11.0}$ (returned as the rational $33/3$) <p>Final Answers: [11.0, 11.0, 11.0] Initial Weights: $w_i^{(0)} = 1/3$ for all i.</p> <p>Controller Signals ($t=0$): the endpoint cluster ($d=1.0$) is unique, but at the early depth $d=0.4$ the three trajectories form three distinct clusters (literal substitution / stepwise variables / symbolic Rational), giving $\bar{H}_{0.4}^{(0)} = 1.0$ while all other depths are merged. After the depth-discounted aggregation in Eq. 5, $\bar{H}^{(0)} = \gamma^{0.6} = 0.81 \geq \tau_H$. By convention $\Delta^{(0)} = 0$.</p> <p>Decision (Stage 1): $\bar{H}^{(0)} \geq \tau_H \Rightarrow \text{REFLECT}$.</p> <hr/> <p>Reflection Step (round $t=1$, top-two endpoint clusters C_A, C_B): since the pool has a single endpoint cluster, the controller falls back to the two highest-weight representatives within it: $y_A = y_1$, $y_B = y_3$. The arbiter is queried with the prompt of Appendix D:</p> <p><i>Errors in y_A:</i> “None. Direct substitution is computationally correct: $(16 + 50)/6 = 11$.”</p> <p><i>Errors in y_B:</i> “No semantic error: returns the value as a Rational object rather than a numeric scalar, but the underlying value $33/3 = 11$ is identical.”</p> <p><i>Refined Solution y_{new}:</i> “Both candidates yield the same value. Adopt the simpler form: $\frac{x^4+2y^2}{6} = \frac{2^4+2\cdot5^2}{6} = \frac{16+50}{6} = \frac{66}{6} = 11$. Agrees with both A and B.”</p> <p>LRP Update (Eq. 10): the arbiter ratifies the dominant endpoint cluster, so y_1, y_2, y_3 all inherit $\mathcal{S}^+$ and y_{new} enters with $w_{\text{new}}^{(1)} = 1/4$ and $\mathcal{S}^+$. After the reward update and renormalization: $w^{(1)} \approx [0.245, 0.245, 0.245, 0.265]$.</p> <p>Controller Signals ($t=1$): the L1 symbolic check merges the Rational and float variants, so the endpoint depth and the intermediate depths all collapse to a single cluster; only $d=0.2$ retains a small residual split between the literal-substitution setup and the others ($\bar{H}_{0.2}^{(1)} = 0.24$), leaving $\bar{H}^{(1)} = 0.18 < \tau_H$. The endpoint cluster distribution is unchanged across $t=0 \rightarrow t=1$ (a single cluster holding all mass), so $\Delta^{(1)} = 0.00 < \tau_\Delta$.</p> <p>Decision (Stage 2): both signals fall below their thresholds $\Rightarrow \text{TERMINATE}$.</p>	<p>Round 1 11.0 (✓)</p>

Table 11: A MATH-500 case (GPT-4o). Three perspectives agree on the final answer but diverge in their intermediate representations, which the depth-aggregated Reflecting Entropy detects ($\bar{H}^{(0)}=0.81 \geq \tau_H$) while a surface-level vote would not. A single arbiter call confirms the dominant value, LRP redistributes weight, and the Stage-2 check terminates after round $t=1$, well within $T_{\max}=4$.

Iterative Reflection Process	Output
<p>Question: The numbers 1447, 1005 and 1231 have something in common: each is a 4-digit number beginning with 1 that has exactly two identical digits. How many such numbers are there?</p> <p>Generated Perspectives ($m=3$):</p> <p>I_1 [brute-force enumeration]: “Iterate over all 4-digit numbers starting with 1 and count those whose digit-frequency profile matches the constraint.”</p> <p>I_2 [combinatorial decomposition]: “Decompose by which digit position holds the duplicate; apply the multiplication principle case-by-case.”</p> <p>I_3 [complementary counting]: “Count numbers with all distinct digits and subtract from the total 1000.”</p> <p>Initial Pool $TP^{(0)}$ (PoT executions):</p> <ul style="list-style-type: none"> • y_1 (I_1): <code>count=0; for n in range(1000,2000): if sorted(Counter(str(n)).values())==[1,1,2]: count+=1</code> → 432 • y_2 (I_2): “Two cases: (a) the duplicate is the leading 1; (b) the duplicate is among the last three digits. Cases summed: $9 \cdot 9 \cdot 3 + \binom{3}{2} \cdot 9 \cdot 9 \cdot 1$”; arithmetic slip yields $243 + 81 = \mathbf{324}$ • y_3 (I_3): “Distinct-digit count: $9 \cdot 9 \cdot 8 = 648$; total starting with 1: 1000; <i>exactly two identical</i> mistakenly equated with ‘not all distinct’, giving $1000 - 648 = \mathbf{352}$” <p>Final Answers: [432, 324, 352] Initial Weights: $w_i^{(0)} = 1/3$ for all i.</p> <p>Controller Signals ($t=0$): three distinct endpoint clusters with equal mass give $\bar{H}_{1,0}^{(0)} = 1.0$, and the mid-to-late depths $d \in \{0.6, 0.8\}$ are fragmented in the same way (the three methods diverge as soon as the counting strategy is committed), while the shallow depths $d \in \{0.2, 0.4\}$ still share a common problem restatement. Aggregating over $\mathcal{D}$ gives $\tilde{H}^{(0)} = 2.80 \geq \tau_H$. $\Delta^{(0)} = 0$ by convention. $\Rightarrow$ REFLECT (Stage 1).</p> <hr/> <p>Reflection Step ($t=1$, top-two clusters $C_A = \{y_1\}, C_B = \{y_3\}$):</p> <p><i>Errors in y_A:</i> “None. The brute-force enumeration is exhaustive over the 1000-element domain.”</p> <p><i>Errors in y_B:</i> “Conflates ‘exactly two identical digits’ with ‘not all distinct’; the latter also includes triples $(1,1,1,d)$ and quadruples $(1,1,1,1)$ which must be excluded.”</p> <p><i>Refined Solution y_{new}:</i> “Trust A’s enumeration: 432. Agrees with A.” (answer = 432)</p> <p>LRP Update: $S^+ \leftarrow \{y_1, y_{\text{new}}\}, S^- \leftarrow \{y_3\}$; y_2 receives no tag this round. After Eq. 10 and renormalization, $w^{(1)} \approx [0.41, 0.18, 0.13, 0.28]$.</p> <p>Controller Signals ($t=1$): the endpoint distribution is now $(0.69, 0.18, 0.13)$ over the clusters $\{432\}, \{324\}, \{352\}$, so $\bar{H}_{1,0}^{(1)} = 0.76$ and $\tilde{H}^{(1)} = 2.11 \geq \tau_H$. Relative to the uniform $t=0$ distribution this is a sizeable movement, $\Delta^{(1)} = 0.07$. $\Rightarrow$ REFLECT (Stage 1).</p> <hr/> <p>Reflection Step ($t=2$, top-two clusters $C_A = \{y_1, y_4\}, C_B = \{y_2\}$):</p> <p><i>Errors in y_A:</i> “None.” <i>Errors in y_B:</i> “Case-(a) over-counts: when the duplicate ‘1’ is at the leading position, the second ‘1’ may collide with one of three remaining slots, but the multiplication $9 \cdot 9 \cdot 3$ omits the position-symmetry correction; case-(b) similarly under-counts the $\binom{3}{2}$ patterns where neither duplicate is the leading 1.”</p> <p><i>Refined Solution y_{new}:</i> “A is correct: 432.” (answer = 432)</p> <p>LRP Update: $S^+ \leftarrow C_A \cup \{y_{\text{new}}\}, S^- \leftarrow C_B$; $w^{(2)} \approx [0.39, 0.10, 0.08, 0.20, 0.23]$.</p> <p>Controller Signals ($t=2$): the endpoint distribution tightens to $(0.82, 0.10, 0.08)$, giving $\bar{H}_{1,0}^{(2)} = 0.54$ and $\tilde{H}^{(2)} = 1.52 \geq \tau_H$; $\Delta^{(2)} = 0.01$. $\Rightarrow$ REFLECT (Stage 1).</p> <hr/> <p>Reflection Step ($t=3$, top-two clusters $C_A = \{y_1, y_4, y_5\}, C_B = \{y_2\}$): both representatives now endorse 432; refined solution: “Agrees with A.” (answer = 432)</p> <p>LRP Update: weights $w^{(3)} \approx [0.36, 0.08, 0.06, 0.18, 0.18, 0.14]$.</p> <p>Controller Signals ($t=3$): the endpoint distribution reaches $(0.86, 0.08, 0.06)$ and, crucially, the repeated arbiter passes have also consolidated the intermediate depths onto the enumeration route, so only $d=1.0$ still contributes: $\tilde{H}^{(3)} = 0.46 < \tau_H$. The endpoint distribution has almost stopped moving, $\Delta^{(3)} = 0.002 < \tau_\Delta$. $\Rightarrow$ TERMINATE (Stage 2).</p> <p>Terminal Pool ($N=6$): answers [432, 324, 352, 432, 432, 432], dominant endpoint cluster aggregates $\sum_{y_i \in C_A} w_i^{(3)} \approx 0.86$.</p> <p>Controller Trajectory: $\tilde{H}^{(t)} : 2.80 \rightarrow 2.11 \rightarrow 1.52 \rightarrow 0.46$ – monotone descent within the budget $T_{\max}=4$.</p>	<p>Round 3 432 (✓)</p>

Table 12: An AIME case (GPT-4o). Three perspectives produce three different final answers, so $\tilde{H}^{(t)}$ stays above τ_H for the first three rounds and Stage 1 keeps invoking the arbiter. The arbiter eliminates the combinatorial errors one cluster at a time, while Eq. 10 concentrates belief mass on the consistently endorsed cluster. The Stage-2 check terminates the loop at $t=3$, within the budget $T_{\max}=4$.

Iterative Reflection Process	Output
<p>Question: Would a vegan eat dishes prepared with bone char-filtered sugar?</p> <p>Generated Perspectives ($m=3$):</p> <p>I_1 [ingredient-level reasoning]: “Inspect the final ingredient list and check whether any animal-derived component remains.”</p> <p>I_2 [definitional reasoning]: “Apply the standard dictionary definition of ‘vegan’ (avoids animal products) to the dish in question.”</p> <p>I_3 [analogical reasoning]: “Compare bone-char-filtered sugar to other widely accepted vegan staples (beet sugar, agave) by chemical composition.”</p> <p>Initial Pool $TP^{(0)}$:</p> <ul style="list-style-type: none"> • y_1 (I_1): “Final sugar contains no animal residue $\Rightarrow$ vegan-safe $\Rightarrow$ would eat.” $\rightarrow$ yes • y_2 (I_2): “Sugar is plant-derived; vegans avoid <i>animal products</i>, and sugar is not an animal product $\Rightarrow$ would eat.” $\rightarrow$ yes • y_3 (I_3): “Bone-char-filtered cane sugar is chemically indistinguishable from vegan-safe beet sugar $\Rightarrow$ would eat.” $\rightarrow$ yes <p>Final Answers: [yes, yes, yes] Initial Weights: $w_i^{(0)} = 1/3$ for all i.</p> <p>Controller Signals ($t=0$): the endpoint cluster ($d=1.0$) is unique, so $\bar{H}_{1.0}^{(0)} = 0$: <i>a final-answer vote would see unanimous agreement and detect nothing</i>. At the shallow depths $d \in \{0.2, 0.4, 0.6\}$, however, the three trajectories invoke three different premise frames (ingredient-level, definitional, analogical) and therefore form three equal-mass clusters, giving $\bar{H}_d^{(0)} = 1$ at each of them. Depth aggregation (Eq. 5) yields $\tilde{H}^{(0)} = \gamma^{0.8} + \gamma^{0.6} + \gamma^{0.4} = 2.43 \geq \tau_H$. By convention $\Delta^{(0)} = 0$, but Stage 2 is never reached.</p> <p>Decision (Eq. 8, Stage 1): $\tilde{H}^{(0)} \geq \tau_H \Rightarrow$ REFLECT. Note that the multi-depth signal alone is what triggers reflection here; restricting $\mathcal{D} = \{1.0\}$ would give $\tilde{H}^{(0)} = 0$ and terminate immediately on a wrong unanimous answer.</p> <hr/> <p>Reflection Step ($t=1$, top-two representatives $y_A = y_1, y_B = y_2$ from the single endpoint cluster):</p> <p><i>Errors in y_A:</i> “Collapses the meaning of ‘vegan’ to ingredient-level reasoning; the standard definition explicitly includes animal-derived processing aids.”</p> <p><i>Errors in y_B:</i> “Equivocates on ‘animal product’: bone char <i>is</i> an animal-derived processing aid, and major vegan-certifying bodies (e.g., The Vegan Society, PETA) classify bone-char-filtered sugar as non-vegan.”</p> <p><i>Refined Solution y_{new}:</i> “Vegan ethics extends to animal-derived processing aids, not just final ingredients. A strict vegan would therefore <i>not</i> eat dishes prepared with bone-char-filtered sugar. New third answer: no.” (answer = False)</p> <p>LRP Update: $\mathcal{S}^+ \leftarrow \{y_{\text{new}}\}, \mathcal{S}^- \leftarrow \{y_1, y_2\}$; y_3 receives no tag this round. After Eq. 10, $w^{(1)} \approx [0.13, 0.13, 0.27, 0.47]$.</p> <p>Controller Signals ($t=1$): a new endpoint cluster (no) appears alongside the original (yes), so the endpoint depth now contributes $\bar{H}_{1.0}^{(1)} \approx 1.00$ and $\tilde{H}^{(1)} = 2.69 \geq \tau_H$. The endpoint distribution shifts from (1.0, 0) at $t=0$ to (0.53, 0.47) at $t=1$, producing $\Delta^{(1)} = 0.20 \geq \tau_\Delta \Rightarrow$ REFLECT (Stage 1).</p> <p>Reflection Step ($t=2$, top-two clusters $C_A = \{y_1, y_2, y_3\}, C_B = \{y_{\text{new}}\}$): the arbiter re-examines the original three trajectories against y_{new} and concludes that the bone-char ethics argument is decisive; refined solution agrees with y_B (answer = False).</p> <p>LRP Update: $\mathcal{S}^+ \leftarrow C_B \cup \{y_{\text{new}}\}, \mathcal{S}^- \leftarrow C_A$; $w^{(2)} \approx [0.05, 0.05, 0.09, 0.46, 0.35]$.</p> <p>Controller Signals ($t=2$): two arbiter passes have consolidated the premise frames, so the shallow depths no longer contribute and only the endpoint depth remains: the distribution tightens from (0.53, 0.47) to (0.19, 0.81), giving $\bar{H}_{1.0}^{(2)} = 0.69$ and hence $\tilde{H}^{(2)} = 0.69 \geq \tau_H$, while the movement is already modest, $\Delta^{(2)} = 0.07 < \tau_\Delta \Rightarrow$ REFLECT (Stage 1): the pool leans towards no but has not yet converged tightly enough for Stage 2 to be reached.</p> <p>Reflection Step ($t=3$, top-two clusters $C_A = \{y_{\text{new}}, y_{\text{new}}\}, C_B = \{y_1, y_2, y_3\}$): both leading representatives now endorse no; the arbiter confirms the processing-aid argument once more (answer = False).</p> <p>LRP Update: $\mathcal{S}^+ \leftarrow C_A \cup \{y_{\text{new}}\}, \mathcal{S}^- \leftarrow C_B$; $w^{(3)} \approx [0.01, 0.01, 0.03, 0.37, 0.32, 0.26]$.</p> <p>Controller Signals ($t=3$): the endpoint distribution reaches (0.05, 0.95), so $\bar{H}_{1.0}^{(3)} = 0.30$ and $\tilde{H}^{(3)} = 0.30 < \tau_H$; the distribution has almost stopped moving, $\Delta^{(3)} = 0.02 < \tau_\Delta \Rightarrow$ TERMINATE (Stage 2).</p> <p>Terminal Pool ($N=6$): the dominant endpoint cluster (no) aggregates 0.95 of the belief mass, within the budget $T_{\text{max}}=4$.</p>	<p>Round 3 no (✓)</p>

Table 13: A StrategyQA-style case (GPT-4o) demonstrating why Reflecting Entropy aggregates over *multiple depths*. All three perspectives reach the same surface answer (yes), so the endpoint depth is fully converged ($\bar{H}_{1.0}^{(0)} = 0$) and a majority vote or any endpoint-only signal would terminate immediately on this *wrong* unanimous answer. The three trajectories nevertheless argue from incompatible premise frames, which the shallow depths $d \in \{0.2, 0.4, 0.6\}$ expose, driving $\tilde{H}^{(0)} = 2.43 \geq \tau_H$ and triggering the reflection that recovers the correct answer at $t=3$.