WHEN DOES SELF-EVALUATION HELP IN REASONING? AN ANATOMY OF TOT ON MATHEMATICAL TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

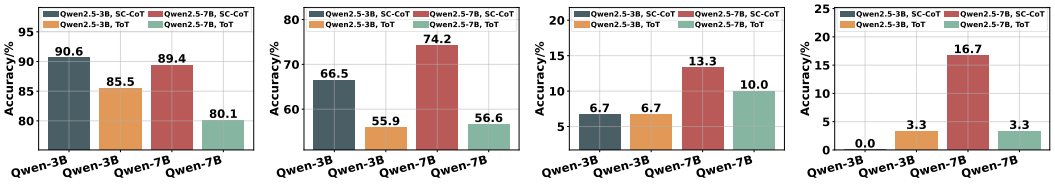
Step-wise reasoning, as a fundamental capability of large language models (LLMs), has significantly advanced complex reasoning through decomposing a problem into a sequence of coherent intermediate thoughts conditioned on context information and prior knowledge. As two representatives, sampling-based reasoning paradigms (e.g., Self-Consistency Chain-of-Thought, SC-CoT) achieve impressive performance on tasks with deterministic answers by aggregating multiple trajectories via majority voting, while search-based reasoning paradigms (e.g., Tree-of-Thoughts, ToT) excel on search-intensive tasks that require explicit step-wise evaluations. However, despite explicit step-wise evaluations, ToT often falls short in mathematical reasoning tasks. Rather than simply attributing this problem to unreliable self-evaluations, in this paper, we revisit ToT and explore the properties of its evaluation behaviors from both theoretical and empirical perspectives. Different from SC-CoT, which marginalizes out thoughts and benefits from a concentrated distribution around the correct answer (i.e., low answer diversity), the performance of ToT is correlated with the coverage of branches in a given budget and the distributions of self-evaluations. Motivated by these findings, we propose a tool, entropy-guided adaptive reasoning tree (EGART), to further investigate whether evaluations can provide effective signals to select the promising branches. Overall, our findings uncover the behaviors of ToT in reasoning tasks and provide an insight into why explicit self-evaluation steps fail in search-based stepwise reasoning.

1 INTRODUCTION

Large language models (LLMs) (Achiam et al., 2023; Liu et al., 2024; Yang et al., 2024a; Touvron et al., 2023) have demonstrated remarkable capabilities across a wide range of tasks, ranging from practical problem solving (Lewkowycz et al., 2022; Li et al., 2024b) to task planning and collaboration (Deng et al., 2024; Valmeekam et al., 2023; Du et al., 2024; Tian et al., 2026). Reasoning (Bowman et al., 2015; Wei et al., 2022b), as the cornerstone of LLM reasoning, plays an essential role in these impressive achievements. However, the capability of conventional reasoning is constrained by a lack of demonstrations and a limited computational budget for unfolding intermediate reasoning, thereby usually failing to handle complex problems. In this context, test-time scaling has emerged as a promising paradigm to improve the reasoning performance of LLMs with more computational budgets. On the input side, in-context learning (ICL) (Brown et al., 2020) augments prompts with abundant demonstrations as in-context information to have LLMs address complex reasoning by referring to similar counterparts. On the output side, stepwise reasoning, such as Chain-of-Thought (CoT) (Wei et al., 2022b), decomposes the solution into a sequence of coherent intermediate thoughts conditioned on context information and prior knowledge, trading additional computation for better performance. So far, mainstream stepwise reasoning methods can be classified as either the sampling-based paradigm, such as Self-Consistency Chain-of-Thought (SC-CoT (Wang et al., 2023)), or the search-based paradigm, such as Tree-of-Thoughts (ToT (Yao et al., 2023)).

As the representative methods of the two aforementioned reasoning paradigms (Gui et al., 2024; Wang et al., 2023; Yao et al., 2023), SC-CoT and ToT enhance the reasoning capability of LLMs from different perspectives. Specifically, SC-CoT, as a sampling-based paradigm, counteracts potential incorrect reasoning paths by majority voting among multiple sampled responses. In this context, two assumptions are implicitly adopted in SC-CoT: (i) the probability of LLMs generating correct answers to queries is higher than that of making mistakes, and (ii) the answers are deterministic and verifiable. In contrast, ToT, as a search-based paradigm, searches for the optimal intermediate thought

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



(a) GSM8K (b) MATH (c) AIME24 (d) AIME25

Figure 1: **Comparison of SC-CoT and ToT on Mathematical Reasoning Tasks.** We compare the performance of SC-CoT and ToT on mathematical reasoning tasks. According to the empirical results in the figure, we can observe that SC-CoT can achieve better performance than ToT in most cases.

with explicit step-wise evaluation based on a weaker assumption that there exist identifiable correct candidates among the sampled intermediate thoughts. Thus, given the differences in underlying mechanisms, the two paradigms exhibit complementary strengths across different tasks. Specifically, SC-CoT achieves better performance on mathematical problems, where the deterministic answers can be aggregated and aligned by marginalizing out the intermediate thoughts, while ToT excels at combinatorial and open-ended tasks, where planning and exploration over intermediate thoughts are required. However, as empirically demonstrated in Fig. 1, ToT typically falls short in mathematical problems. Thus, a question raised here is: *Why does ToT fail to handle typical mathematical problems even though explicit self-evaluations are employed?* In this paper, rather than simply attributing this gap to unreliable evaluations, we revisit ToT with different self-evaluation and branch selection strategies and propose to explore the properties of self-evaluations in LLMs on mathematical tasks.

As an exploration, we provide comprehensive analyses on both SC-CoT and ToT. Theoretically, we find that SC-CoT benefits from a concentrated distribution around the correct answer. Intuitively, a diverse answer distribution, where the probabilities of generating correct and incorrect answers are similar, disperses probability mass across many candidates, making it harder to infer the correct answer. Thus, with the two implicit assumptions above, increasing the sampling size in SC-CoT approaches the true distribution, where the empirical answer distribution concentrates around the correct answer, thereby contributing to successful reasoning. In ToT, given the evaluation is unreliable, the margin between the best correct intermediate thought and the least favored candidate matters so that the correct thought can be selected. Empirically, we demonstrate that enlarging the coverage of intermediate thoughts (e.g., with either non-overlap sampling strategies or independent evaluation strategies) facilitates improving the margin and further enhancing the reasoning performance. In addition, our empirical results further find that self-evaluation signals are meaningful when adopted as high-level signals (e.g., ranking or entropy), instead of just fine-grained magnitude-level signals.

Motivated by these findings, we propose to further probe the effect of intermediate thought coverage from the perspective of evaluation entropy. To this end, we propose an intervention tool, *entropy-guided adaptive reasoning tree* (EGART) (cf. Fig. 5), to adaptively determine the size of preserved intermediate thoughts via the self-evaluation entropy. Intuitively, EGART tends to preserve fewer candidates when the evaluation entropy is low, while it preserves more candidates when the evaluation entropy is high. Along with selection strategies, which either leverage ranking or magnitude information, we can probe the performance changes of ToT. Quantitative results show that a combination of ranking and the entropy of independent evaluations helps achieve comparable performance with fewer intermediate thoughts. Moreover, ranking reveals its stability under unreliable evaluations.

Contributions. Overall, our contributions in this paper can be summarized as the following aspects:

- We find that ToT, which is equipped with explicit self-evaluation, suffers from inferior performance on mathematical tasks compared to sampling-based reasoning paradigms (e.g., SC-CoT) (cf. Fig. 1).
- In Section 2 & 3, we conduct both theoretical and empirical investigations on SC-CoT and ToT to explore their underlying mechanisms in reasoning. Theoretically, SC-CoT benefits from a concentrated distribution around the correct answer, while ToT benefits from an appropriate preservation budget that can cover the identifiable correct answers. Empirically, we find the coverage over candidate intermediate thoughts, instead of evaluation, matters for the reasoning performance of ToT. In addition, we also find that self-evaluation signals can provide meaningful information when they are adopted as high-level signals, such as ranking or evaluation entropy.
- In Section 4, we further probe the effect of coverage with an intervention tool, *entropy-guided adaptive reasoning tree* (EGART). Empirical results reveal that the combination of evaluation entropy and ranking information helps preserve meaningful candidates and retain the performance.

2 ANALYSES OF STEPWISE REASONING PARADIGMS

In this section, we aim to explore the reasoning behaviors of two representative stepwise reasoning paradigms: *Self-Consistency Chain-of-Thought* (i.e., SC-CoT) and *Tree-of-Thoughts* (i.e., ToT). Specifically, we would like to answer two questions: (1) Where are the capability boundaries of the two reasoning paradigms? (2) How do the two paradigms behave in mathematical reasoning tasks?

2.1 PROBLEM FORMULATION OF SC-COT AND TOT

Consider a pre-trained LLM that is parameterized with θ. Let x denote an input composed of necessary instructions and a query, and y denote the ground truth (e.g., y is a numerical answer in mathematical reasoning). In the context of stepwise reasoning, the final result is derived from a sequence of intermediate reasoning steps: $z_1, z_2, \ldots, z_{N_\mathrm{s}-1}$, where $z_{i+1} \sim p(\cdot|x, z_1, z_2, \ldots, z_i; \theta)$ and N_s denotes the total number of steps in reasoning. In this way, we can then generally formulate the reasoning as $\hat{y} = f_\mathrm{e}(z_{N_\mathrm{s}})$, where $z_{N_\mathrm{s}} \sim p(\cdot|x, z_1, \ldots, z_{N_\mathrm{s}-1}; \theta)$ denotes the final reasoning thought, f_e denotes a function for answer extraction, and $\hat{y}$ denotes the answer inferred by the LLM.

SC-CoT. SC-CoT (Wang et al., 2023) technically enhances the naive CoT by performing plurality voting on multiple reasoning trajectories sampled from the LLM. Specifically, SC-CoT first independently samples a set of reasoning paths $\{z_1^{(i)}, z_2^{(i)}, \ldots, z_{N_\mathrm{s}-1}^{(i)}, z_{N_\mathrm{s}}^{(i)}\}_{i=1}^{N_\mathrm{T}}$ from the distribution determined by the given input and model parameters. Let $\mathcal{A} = \{f_\mathrm{e}(z_{N_\mathrm{s}})|z_{1:N_\mathrm{s}} \in \mathrm{supp}\left(p(\cdot|x;\theta)\right)\}$ denote the set of all possible answers. The final result is defined as the most consistent answer in $\mathcal{A}$:

$$\hat{y}_\mathrm{sc} := \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{N_\mathrm{T}} \mathbb{1}\left(\hat{y}^{(i)} = a\right), \tag{1}$$

where $\hat{y}^{(i)} = f_\mathrm{e}(z_{N_\mathrm{s}}^{(i)})$ denotes the answer extracted from the i-th trajectories sampled from the LLM.

ToT. ToT (Yao et al., 2023) formulates the stepwise reasoning into a tree search problem. ToT first decomposes the reasoning into several intermediate steps according to the given problem. At each step, several candidate thoughts are proposed and then evaluated to select the most promising candidates for the next step. Specifically, at the i-th step, suppose there are N_p active states $\{s_n^{(i)}\}_{n=1}^{N_\mathrm{p}}$. For each active state $s_n^{(i)} = [x, z_n^{(1)}, \ldots, z_n^{(i-1)}]$, a set of N_b candidate thoughts $\{z_{n,k}^{(i)}\}_{k=1}^{N_\mathrm{b}} \sim p(\cdot|s_n^{(i)}; \theta)$ is proposed by the LLM. Then, a parametric evaluator (e.g., the LLM itself or a reward model) V_ψ, where ψ denotes the parameters, is applied to select N_p potential solutions from all branches for the next reasoning step. In the final step of each reasoning process, with the set of candidate terminal states $\mathcal{S}_{N_\mathrm{s}}$, the answer is obtained by selecting the state that the evaluator assigns the highest score.

$$\hat{y}_\mathrm{tot} = f_\mathrm{e}(s^*), \text{where } s^* = \arg\max_{s \in \mathcal{S}_{N_\mathrm{s}}} V_\psi\left(s\right). \tag{2}$$

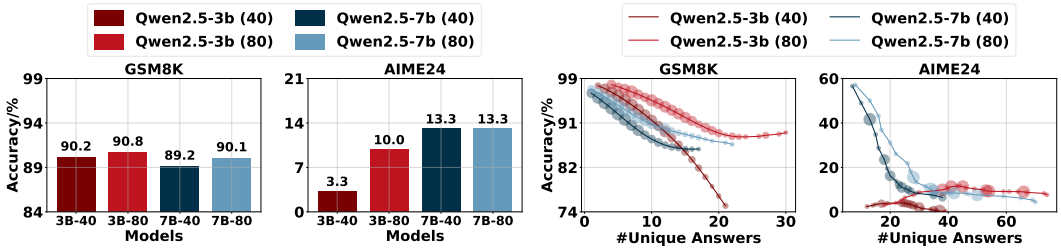
Intuitively, according to the problem formulations above, SC-CoT benefits from the concentrated distribution around the correct answer by abundant sampling, while ToT focuses on exploring the most promising solution by meticulously examining the intermediate solutions at each reasoning step.

2.2 PROBING REASONING CAPABILITY BOUNDARIES OF SC-COT AND TOT

In this subsection, we provide theoretical analyses to probe the capability boundaries of the aforementioned two reasoning paradigms. Specifically, we specify the reasoning capability of the reasoning paradigms as the probability that the paradigms can successfully infer the answer to the given input.

Definition 2.1. *Let $q(a) := P\left(\hat{y} = a|x;\theta\right), a \in \mathcal{A}$, denote the answer distribution over the candidate answer set $\mathcal{A}$ induced by the trajectory distribution. Define the effective answer set as $\mathcal{A}_\theta := \{a \in \mathcal{A} : q(a) > 0\} \subseteq \mathcal{A}$. Let $\{\hat{y}^{(i)}\}_{i=1}^{N_\mathrm{T}}$ denote the answers extracted from N_T i.i.d. sampled reasoning trajectories. Define $N_\mathrm{ans} := \sum_{i=1}^{N_\mathrm{T}} \mathbb{1}\left(\hat{y}^{(i)} = y\right)$ denote the number of reasoning trajectories with the correct answer, and let $N_\mathrm{err} := \max_{a \in \mathcal{A}_\theta \setminus \{y\}} \sum_{i=1}^{N_\mathrm{T}} \mathbb{1}\left(\hat{y}^{(i)} = a\right)$ denote the maximum vote count received by any erroneous answer. The reasoning is deemed successful if and only if $N_\mathrm{ans} > N_\mathrm{err}$.*

Theorem 2.2. **(SC-CoT)** *Consider an effective answer set $\mathcal{A}_\theta$ ($|\mathcal{A}_\theta| < \infty$). Define the probability of LLMs generating correct answers as $p_y := q(y)$ and the probability of generating any erroneous*



(a) Performance on GSM8K and AIME24 (b) Uniqueness of Answers on GSM8K and AIME24

Figure 2: **Empirical Studies on SC-CoT.** (a) The performance of SC-CoT with different LLMs (i.e., Qwen2.5-3B/7B) on GSM8K and AIME24 with 40 and 80 sampling sizes, respectively. (b) Studies on the effect of answer diversity on the performance of different LLMs with different sampling sizes.

answer as $q_a, \forall a \in \mathcal{A}_\theta \backslash \{y\}$, *so that* $\sum_{a \in \mathcal{A}_\theta \backslash \{y\}} q_a = 1 - p_y$. *Define* $q_{\max} := \max_{a \in \mathcal{A}_\theta \backslash \{y\}} q_a$. *Then, the probability that SC-CoT successfully infers the correct answer is bounded as follows:*

$$P(N_{\text{ans}} > N_{\text{err}}) \begin{cases} \leq \exp\left(-\frac{1}{2} N_{\text{T}} \left(p_y - q_{\max}\right)^2\right), & p_y \leq q_{\max}, \\ \geq 1 - \sum_{a \in \mathcal{A}_\theta \backslash \{y\}} \exp\left(-\frac{1}{2} N_{\text{T}} \left(p_y - q_a\right)^2\right), & p_y > q_{\max}. \end{cases}$$

The theoretical results in Theorem 2.2 indicate that the performance of SC-CoT is mainly determined by two aspects: the sampling size N_{T} and the margin of probabilities $p_y - q_{\max}$ over the correct answer and the most competitive erroneous answer. Specifically, from the perspective of the sampling size, in the case where SC-CoT probably generates the correct answer (i.e., $p_y > q_{\max}$), increasing the sampling size (i.e., $N_{\text{T}} \uparrow$) facilitates improving the performance of SC-CoT, since the empirical answer distribution concentrates around the correct answer through abundant sampling. Moreover, given a fixed probability mass $1 - p_y$, a larger effective answer set (i.e., $|\mathcal{A}_\theta| \uparrow$) tends to disperse the voting on erroneous answers (i.e., $q_{\max} \downarrow$) further benefit the performance. On the other hand, in the case where SC-CoT can hardly generate the correct answer (i.e., $p_y < q_{\max}$), simply increasing the sampling size tends to result in the collapse of reasoning performance since the empirical distribution concentrates around those erroneous answers. These observations indicate that SC-CoT benefits from abundant sampling only when LLMs already assign the largest probability mass to the correct answer.

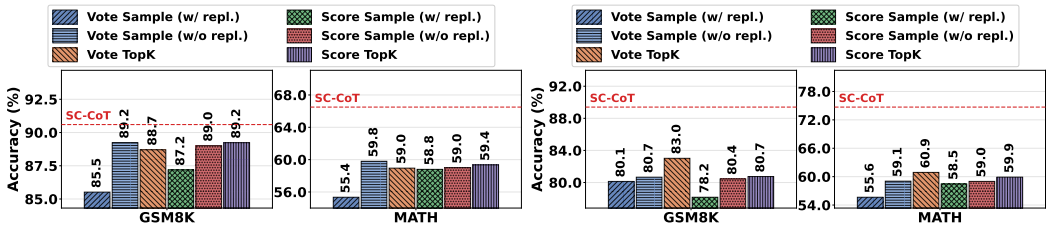
Next, we consider the reasoning capability of ToT. ToT can be treated as a special version of SC-CoT. Specifically, in ToT, reasoning is performed step-by-step explicitly, and explicit evaluations are also employed in each reasoning step for thought selection. Detailed analyses are provided as follows.

Assumption 2.3. *At the* i*-th reasoning step in ToT, given a set of active states obtained from the last step* $\mathcal{S}_{i-1} = \{s_n^{(i-1)}\}_{n=1}^{N_{\text{p}}}$, *we assume that a set of* N_{b} *intermediate thoughts is independently sampled for each single active state from the distribution* $p(\cdot|x, s_n^{(i-1)}; \theta)$. *Let* $\mathcal{Z}_i$ *denote the candidate intermediate thought pool for an active state. Let* $\mathcal{C}_i^+$ *denote the correct thought candidates, and* $\mathcal{C}_i^- = \mathcal{Z}_i \backslash \mathcal{C}_i^+$ *denote the incorrect thought candidates. Given the sampled candidates* $\mathcal{Z}_i$, N_{p} *intermediate thoughts are preserved as the active states for the next step via an evaluation operation.*

Theorem 2.4. (**Single-step ToT**) *Given a set of intermediate thoughts* $\mathcal{Z}_i$ *of the* i*-th step in ToT reasoning, where at least one correct intermediate thought is included.* N_{v} *evaluations are independently conducted for each intermediate thought* $z \in \mathcal{Z}_i$ *and a score* $v_z^{(k)} \in [0, 1]$ *is obtained at the* k*-th evaluation. Let* $\phi_{i,z^+} = \mathbb{E}[v_{z^+}^{(k)}]$ *for* $z^+ \in \mathcal{C}_i^+$ *and* $\rho_{i,z^-} = \mathbb{E}[v_{z^-}^{(k)}]$ *for* $z^- \in \mathcal{C}_i^-$ *denote the expected scores that a correct and an incorrect intermediate thought is selected. Let* $\phi_i = \max_{z \in \mathcal{C}_i^+} \phi_{i,z}$ *obtained at* z_i^+ *and* $\Delta_i = \phi_i - \rho_i^{(N_{\text{p}})}$ *denote the margin between the best correct intermediate thought and the* N_{p}*-th most favored incorrect intermediate thought. With the mild assumptions in Assumption 2.3, the probability of ToT successfully selecting a correct candidate thought is bounded*

$$P\left(E_i\right) \begin{cases} \leq \min\left\{1, |\mathcal{Z}_i| N_{\text{p}} \exp\left(-\frac{1}{2} N_{\text{v}} \Delta_i^2\right)\right\}, & \Delta_i < 0, \\ \geq 1 - \left(|\mathcal{Z}_i| - 1\right) \exp\left(-\frac{1}{2} N_{\text{v}} \Delta_i^2\right), & \Delta_i > 0, \end{cases}$$

where E_i *denotes the event that at least one selected intermediate thought is correct given the set* $\mathcal{Z}_i$.



(a) Acc. of Different Strategies with Qwen2.5-3B (b) Acc. of Different Strategies with Qwen2.5-7B

Figure 3: **Empirical Study on Different Evaluation and Selection Strategies in ToT (on GSM8K and MATH). (a)** When equipped with sampling without replacement, the performance of ToT with voting is significantly improved in almost all cases. Moreover, when equipped with scoring, the performance of ToT with sampling with replacement is also improved. These observations highlight the importance of coverage of intermediate thought in the reasoning of ToT. **(b)** Compared to sampling, ToT with top-k usually achieves better performance, which implies that self-evaluation contains meaningful information when adopted as a high-level signal (e.g., ranking), instead of a fine-grained signal (e.g., magnitude). More detailed discussions are available in Section 3.2 and 3.3.

Theorem 2.4 characterizes the evaluation and selection behaviors in a single ToT reasoning step. We characterize the capability of ToT in identifying the correct intermediate thought by measuring the margin between the best correct intermediate thought and the N_{p}-th most favored incorrect intermediate thought (i.e., Δ_i). Obviously, the bounds obtained above are governed by the margin Δ_i and further amplified by the evaluation budget N_{v}. Specifically, when $\Delta_i > 0$, the probability of selecting the correct intermediate thoughts approaches one with the margin and evaluation budget increasing, whereas when $\Delta_i < 0$, the identification capability collapses significantly. Thus, echoing SC-CoT, the successful reasoning of ToT refers to two aspects: (i) LLMs can correctly evaluate intermediate thoughts, and (ii) correct thoughts are within the coverage of the preservation budget.

3 Empirical Study: Reasoning Behaviors of SC-CoT and ToT

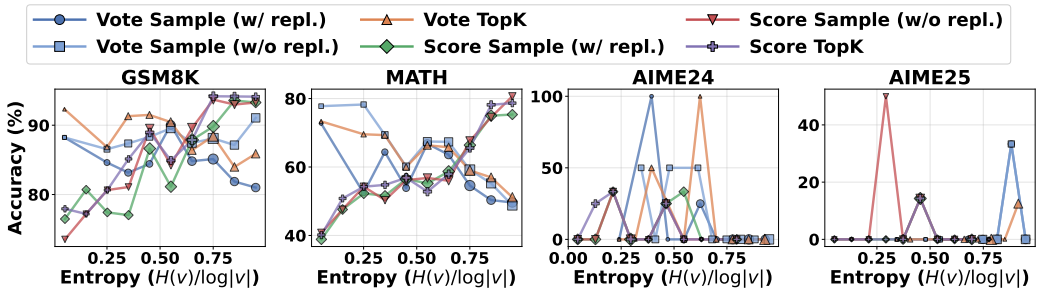
In this subsection, we conduct extensive empirical experiments on two open-source LLMs, Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024a), to validate our theoretical results obtained in Section 2.2 and further explore the potential factors that may be correlated with the performance gap between SC-CoT and ToT on reasoning tasks. Experimental details are available in Appendix C.

3.1 SC-CoT Benefits from Abundant Sampling and Answer Concentration

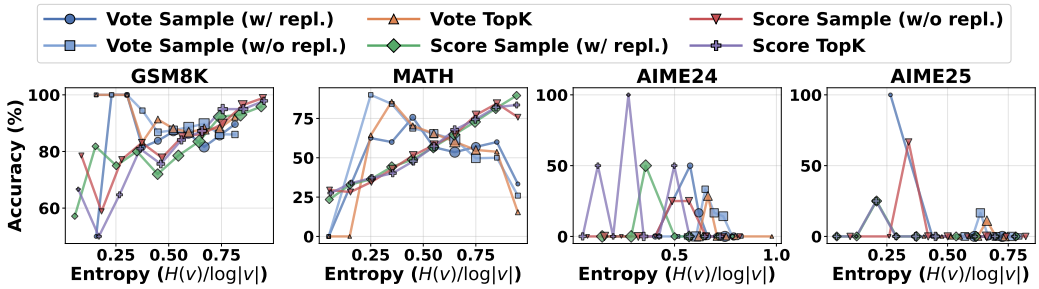
According to Theorem 2.2, in the case that SC-CoT can handle, a concentrated distribution around the correct answer facilitates SC-CoT achieving better performance. As an empirical validation, we perform SC-CoT with two sampling sizes of 40 and 80, respectively, on GSM8K (Cobbe et al., 2021) and AIME24 (Zhang & Math-AI, 2024) benchmarks. The results are visualized in Fig. 2. Two observations should be noted here. **Observation 1.** *The reasoning performance is bootstrapped as the sampling size increases.* Specifically, in Fig. 2(a), the reasoning performance of both LLMs is improved in most cases of GSM8K when the sampling size increases from 40 to 80. However, in hard tasks (e.g., AIME), such a phenomenon is not stable since the two LLMs can hardly sample correct trajectories given the hard problems. **Observation 2.** *A concentrated distribution around answers is positively related to better performance.* In Fig. 2(b), we examine how reasoning performance reacts to different levels of uniqueness of answers. Consistent with our theoretical results, the good performance of SC-CoT is closely related to a concentration distribution of the correct answer. Specifically, in GSM8K, the sampled answers in most cases concentrate on only a few options and higher accuracy is achieved. In AIME24, although the cases with more unique answers tend to achieve better performance, the sampled answers in most cases are diverse, showing inferior performance.

3.2 Coverage, instead of Evaluation, Matters for Reasoning of ToT

In Theorem 2.4, we demonstrate that the core of ToT successfully identifying the correct intermediate thought refers to the capability of LLMs in accurate evaluation and the ranking of correct intermediate thoughts among all candidates. Due to the differences between the intermediate thoughts and alignable answers, in this section, we explore the reasoning behaviors of ToT via a series of experiments on evaluation and selection strategies adopted in ToT. Specifically, we consider two evaluation strategies: voting and scoring. Voting is performed by feeding all intermediate thoughts to the LLM and having



(a) Entropy of the First Evaluation Round on All Benchmarks with Qwen2.5-3B



(b) Entropy of the Second Evaluation Round on All Benchmarks with Qwen2.5-3B

Figure 4: **Visualization Results of the Effect of Evaluation Entropy on the Reasoning Performance of ToT (Qwen2.5-3B).** We explore the correlation between the reasoning performance of ToT and the evaluation entropy of both two evaluation rounds via the normalized evaluation entropy of the evaluation vector $\boldsymbol{v}$: $H(\boldsymbol{v})/\log(|\boldsymbol{v}|)$. Generally, in the case of voting, lower evaluation entropy is correlated to better performance, whereas in the case of scoring, higher evaluation entropy corresponds to better performance (cf. results in GSM8K and MATH). However, in hard tasks (e.g., AIME benchmarks), such a phenomenon is not observed. More results of Qwen2.5-7B are available in Fig. 8. The observations in Qwen2.5-7B are consistent with those shown in the Qwen2.5-3B case.

it select an optimal one, while scoring evaluates thoughts one by one with a specific score. In addition, we also consider different selection strategies, including sampling with and without replacement.

Evaluation Is Insufficiently Reliable; Coverage Matters. We first examine the reliability of self-evaluation. Specifically, we compared the performance of ToT with two different selection strategies: *sampling with replacement* and *sampling without replacement*. The intuition is that sampling with replacement tends to repeatedly select the candidates with higher scores. If self-evaluation is reliable, sampling with replacement will achieve better performance. Results are visualized in Figs. 3 and 7. According to the comparison, sampling without replacement achieves significantly better performance than sampling with replacement in almost all cases. This phenomenon implies that the best correct intermediate thoughts are probably with relatively lower scores, which is consistent with our analyses in Theorem 2.4. Thus, self-evaluation is empirically not reliable enough. A wide coverage, which contains as many correct intermediate thoughts as possible, matters. Theoretically, a wide coverage corresponds to a larger margin in Theorem 2.4, which helps preserve correct intermediate thoughts.

Independent Evaluation Works Better than Comparison. According to the results, we also notice that ToT with scoring performs better in most cases than ToT with voting under the setting of sampling with replacement. From the perspective of mechanism, voting aims to select an optimal branch via a comparison among all candidates, while scoring obtains an evaluation distribution based on branch-wise independent evaluations. Thus, the evaluation distribution of voting tends to be sharper than that of scoring. In this case, the smooth evaluation distribution of scoring contributes to enlarging the coverage of intermediate thoughts, which facilitates achieving better reasoning performance.

3.3 Are Self-Evaluation Signals Completely Not Trustworthy?

The empirical studies above raise a question: *Are self-evaluation signals completely not trustworthy?* In Figs. 3 and 7, we also validate the performance of ToT that is equipped with top-k for intermediate thought selection. An interesting phenomenon is that ToT with top-k usually achieves better performance than that with sampling without replacement, especially on larger models (e.g., Qwen2.5-7B). This phenomenon seems to support an opposite argument: the self-evaluation signals can reflect the

quality of intermediate thoughts and help select the correct intermediate thoughts. Intuitively, ranking, as a high-level signal derived from evaluations, is more effective than fine-grained magnitude signals.

Moreover, as evaluation strategies, voting and scoring measure different aspects of intermediate thought generation. Specifically, voting aims to select an optimal intermediate thought from a set of candidates, thereby showing the confidence of LLMs in intermediate thoughts. Scoring evaluates each intermediate thought independently with a specific score, thereby reflecting the general quality of the candidate pool. Analogous to SC-CoT, we empirically study the correlation between the performance of ToT and the concentration distribution of evaluation. We here characterize the concentration of evaluations with the normalized entropy of evaluations $H(\boldsymbol{v})/\log(|\boldsymbol{v}|)$, where $\boldsymbol{v} \in \mathbb{R}^{N_\text{b}}$ denotes an evaluation vector. The results are visualized in Figs. 4 and 8. According to the results, we can observe a clear correlation between the evaluation entropy and the performance of ToT on easier tasks that LLMs can handle, such as GSM8K and MATH. Specifically, when equipped with scoring, the performance of ToT improves as the evaluation entropy increases. In contrast, when equipped with voting, we can observe a trend that ToT can achieve good performance when the evaluation entropy is low. These phenomena are consistent with the properties of the two strategies. A low entropy regarding voting shows that LLMs are more confident, whereas a high entropy regarding scoring reveals that the candidates in the pool are comparably good. Notably, in MATH, we also observe that low entropy regarding voting can also result in performance collapse, indicating that a confident consensus may collectively favor a flawed thought on hard problems. Moreover, we also observe that there exists a position bias in voting (cf. C.2), which helps explain the unstable correlation in figures.

4 A Probe on Coverage from Perspective of Evaluation Entropy

In this section, we propose to validate the two signals derived from self-evaluation. On the one hand, we exploit the evaluation entropy to control the preservation budget. Specifically, more reasoning paths are preserved when the entropy is high while fewer trajectories are preserved when the entropy is low. On the other hand, we consider both top-k and sampling for intermediate thought selection to investigate the coverage capability of ranking information contained in self-evaluation. Through a series of empirical results, we would like to further validate (1) *the behaviors and the reliability of self-evaluation strategies* and (2) *whether ranking helps capture meaningful intermediate thoughts*.

4.1 Entropy-Guided Adaptive Reasoning Tree

In this section, we first propose a tool, *entropy-guided adaptive reasoning tree* (EGART), to further investigate the properties of self-evaluation in ToT. EGART adaptively determines the size of intermediate thoughts preserved for the next step based on the evaluation entropy over all candidates.

Given a set of intermediate thoughts $\mathcal{Z}_i$, N_v evaluations are independently conducted to validate their quality. For voting, due to the position bias (cf. Appendix C.2 and Wang et al. (2024)), a random shuffle should be performed on the intermediate thoughts at the beginning of each vote. Such a phenomenon does not exist in scoring since intermediate thoughts are evaluated one-by-one independently. Then, a one-hot vector $\boldsymbol{v}_r$ for the r-th evaluation is obtained from LLMs. With all N_v independent evaluations, we obtain the final evaluation results by summing all evaluations: $\boldsymbol{v} = \sum_{r=1}^{N_\text{v}} \boldsymbol{v}_r$. By normalizing the evaluation result vector (e.g., $\bar{\boldsymbol{v}} = \boldsymbol{v}/||\boldsymbol{v}||$), we can approximate the distribution of the evaluations of LLMs on the intermediate thoughts. Intuitively, the concentrated distribution indicates high confidence. For simplicity, we here adopt the entropy of evaluation to describe the confidence of the evaluation: $H_\text{v} = -\bar{\boldsymbol{v}}^\top \log \bar{\boldsymbol{v}}$. Thus, given the evaluation regarding the candidates, we can adaptively determine the size of the intermediate thoughts preserved for the next step through $K = \lceil \exp(H_\text{v}/\tau) \rceil$, where $\tau \in \mathbb{R}^+$ is a temperature coefficient for confidence calibration and $\lceil \cdot \rceil$ denotes the ceiling operation.

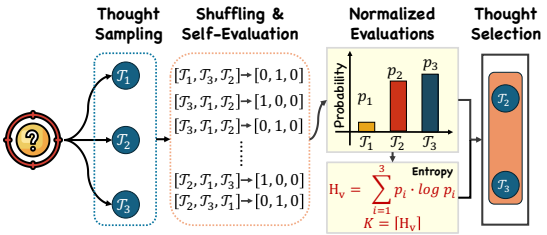


Figure 5: An illustration of the workflow of our proposed entropy-guided adaptive reasoning tree.

Implementation Details. In practical implementations, high evaluation entropy may result in a large K (e.g., when $\tau < 1$) that may be out of the range of all available intermediate thoughts. Thus, in the voting case, we constrain the largest K as the size of all available thoughts, whereas in the scoring case, we set the maximum preservation size for each thought as the default setting in the vanilla ToT.

Table 1: Empirical results of accuracy (Acc.) and the *average* number of reasoning paths (S_{avg}) per query in the final reasoning step on four mathematical reasoning benchmarks, GSM8K, MATH, and AIME 24 & 25, with Qwen2.5-Instruct Models. We highlight the best performance in **bold** and the second best performance with an underline. All reported results are the average of seeds 41 and 42.

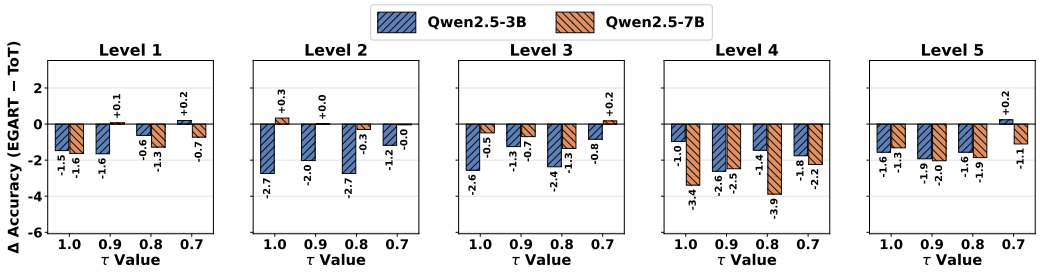
Methods	GSM8K		MATH		AIME 24		AIME 25	
	Acc. (%) ↑	S_{avg} ↓	Acc. (%) ↑	S_{avg} ↓	Acc. (%) ↑	S_{avg} ↓	Acc. (%) ↑	S_{avg} ↓
`Qwen2.5-3B-Instruct`								
CoT	64.1±2.5	1	41.7±1.8	1	5.0±2.4	1	1.7±2.3	1
CoT-SC	**90.8±0.3**	40	**66.7±0.2**	40	8.4±2.3	40	0.0±0.0	40
ToT Score TopK	89.1±0.2	45	59.5±0.1	45	8.4±2.3	45	5.0±2.4	45
EGART Score TopK ($\tau = 1.0$)	87.2±0.2	32.2	58.0±0.7	26.2	6.7±0.0	20.2	1.7±2.3	18.8
EGART Score TopK ($\tau = 0.8$)	87.6±0.1	37.8	58.2±0.4	31.8	8.4±2.3	23.4	5.0±2.4	25.7
EGART Score TopK ($\tau = 0.7$)	88.1±0.1	39.7	59.2±0.0	34.4	5.0±2.4	26.9	**10.0±4.7**	25.9
ToT Score Sample	88.8±0.3	45	59.7±1.0	45	5.0±2.4	45	5.0±2.4	45
EGART Score Sample ($\tau = 1.0$)	87.0±0.8	32.4	57.8±0.2	26.1	5.0±2.4	20.2	3.3±0.0	21.2
EGART Score Sample ($\tau = 0.8$)	88.0±0.5	37.7	58.7±0.9	32.0	**10.0±4.7**	21.8	5.0±7.1	23.6
EGART Score Sample ($\tau = 0.7$)	88.1±0.7	39.9	59.2±0.4	34.7	8.3±7.1	26.8	**10.0±0.0**	26.1
ToT Vote TopK	88.7±0.1	45	60.1±1.5	45	8.4±2.3	45	1.7±2.3	45
EGART Vote TopK ($\tau = 1.0$)	88.1±0.1	32.6	58.2±1.1	35.1	8.4±2.3	35.8	1.7±2.3	35.8
EGART Vote TopK ($\tau = 0.8$)	89.2±0.4	55.1	61.4±0.4	58.9	6.7±4.7	61.8	3.3±0.0	58.3
ToT Vote Sample	88.6±0.3	45	59.2±1.4	45	6.7±0.0	45	0.0±0.0	45
EGART Vote Sample ($\tau = 1.0$)	87.6±0.3	32.5	58.8±0.8	35.1	3.3±0.0	35.5	3.3±0.0	36.9
EGART Vote Sample ($\tau = 0.8$)	88.9±0.1	53.8	60.4±0.6	59.0	6.7±0.0	56.3	5.0±2.4	56.8
`Qwen2.5-7B-Instruct`								
CoT	64.7±0.4	1	57.2±0.3	1	8.4±2.3	1	5.0±2.4	1
CoT-SC	**89.4±0.0**	40	**74.4±0.3**	40	15.0±2.4	40	**18.3±2.4**	40
ToT Score TopK	80.5±0.3	45	60.0±0.1	45	**16.7±4.7**	45	8.4±2.3	45
EGART Score TopK ($\tau = 1.0$)	81.2±0.2	39.9	58.0±0.1	38.2	13.3±0.0	25.0	5.0±2.4	24.0
EGART Score TopK ($\tau = 0.8$)	80.4±0.4	42.9	58.4±0.6	41.5	**16.7±4.7**	32.4	8.4±2.3	29.8
EGART Score TopK ($\tau = 0.7$)	81.4±0.4	43.7	59.5±1.1	42.3	13.3±0.0	36.6	6.7±0.0	34.5
ToT Score Sample	80.8±0.6	45	59.9±0.2	45	10.0±0.0	45	6.7±4.7	45
EGART Score Sample ($\tau = 1.0$)	79.9±0.6	39.9	59.0±0.1	38.3	15.0±7.1	25.6	1.7±2.3	25.7
EGART Score Sample ($\tau = 0.8$)	81.0±0.5	43.0	58.5±0.4	41.4	11.7±2.3	33.5	6.7±0.0	31.6
EGART Score Sample ($\tau = 0.7$)	82.0±0.5	43.5	59.3±0.4	42.3	13.4±9.4	36.7	3.4±4.7	35.4
ToT Vote TopK	82.9±0.1	45	58.8±0.6	45	11.7±2.3	45	3.4±4.7	45
EGART Vote TopK ($\tau = 1.0$)	80.9±1.7	25.1	58.2±0.5	25.5	11.7±2.3	27.2	6.7±0.0	26.2
EGART Vote TopK ($\tau = 0.7$)	82.6±0.7	54.8	59.0±0.4	56.1	10.0±0.0	61.4	3.3±0.0	63.3
ToT Vote Sample	80.2±0.1	45	59.3±0.5	45	11.7±2.3	45	5.0±2.4	45
EGART Vote Sample ($\tau = 1.0$)	80.2±0.0	24.9	57.5±0.4	25.5	11.7±2.3	27.1	5.0±7.1	25.8
EGART Vote Sample ($\tau = 0.7$)	81.0±0.4	54.0	59.7± 1.5	56.3	8.4±2.3	56.5	3.4±4.7	59.0

4.2 Quantitative Results on Self-Evaluation and Ranking

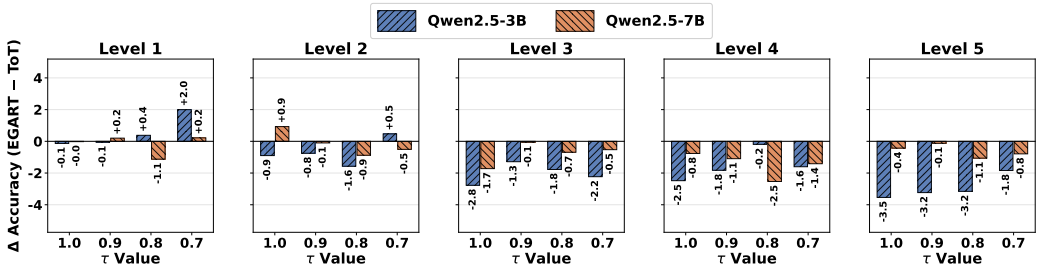
We implement EGART with different self-evaluation and thought selection strategies on two open-source models (e.g., Qwen2.5-3B and Qwen2.5-7B). As a comparison, we consider the following baselines: CoT (Wei et al., 2022b), SC-CoT (Wang et al., 2023), and ToT (Yao et al., 2023). Implementation details are available in Appendix D. Quantitative results are reported in Table 1 & 2.

A Validation on Different Behaviors of Two Evaluation Strategies. As observed in Fig. 4 & 8, in GSM8K and MATH, better performance is correlated with higher entropy of scoring and lower entropy of voting. In Table 1, we observe that more reasoning branches (i.e., S_{avg}) are preserved on simple tasks and fewer branches are preserved on hard tasks (e.g., AIME) when adopting scoring. This phenomenon matches our intuition that candidates with good quality are generated when tasks are simple, while correct intermediate thoughts can hardly be generated when tasks are hard. Meanwhile, voting behaves in an opposite way, matching the preservation strategy controlled by uncertainty.

Scoring Enables More Effective Budget Allocation for Thought Coverage. Compared to the voting counterparts, EGART with scoring achieves comparable performance to baselines with fewer



(a) Differences between EGART Score TopK and ToT baseline with various τ values



(b) Differences between EGART Score Sampling and ToT baseline with various τ value

Figure 6: **An Investigation on Performance Changes of EGART (with scoring) with τ Values on Different Levels of Questions**. We here report the performance changes of EGART Score TopK and EGART Score Sample, respectively, with various τ on different levels of questions in MATH dataset.

reasoning paths, indicating an effective coverage of meaningful intermediate thoughts. For example, in the EGART Score TopK ($\tau = 1.0$) case of Qwen2.5-7B, EGART improves accuracy from 80.5% to 81.2% with $\sim 12\%$ fewer reasoning paths. Moreover, on AIME24, EGART Score TopK ($\tau = 0.8$) matches the performance of the baseline with 72% the budget of the baseline. In contrast, EGART with voting as the evaluation strategy usually achieves better performance with more reasoning paths.

Ranking Remains Stable in Voting. When adopting scoring for self-evaluation, neither top-k nor sampling exhibits a consistent advantage across models and benchmarks. However, when voting is adopted as the evaluation strategy, top-k achieves better performance than sampling even with a low preservation budget (e.g., $\tau = 1.0$), particularly in GSM8K and MATH. This observation demonstrates that the ranking information of voting remains meaningful for covering promising intermediate thoughts, even when their fine-grained distribution is insufficiently reliable for sampling. Nevertheless, such an advantage becomes less stable on the challenging benchmarks (e.g., AIME).

4.3 Qualitative Analyses on Ranking via Different Levels of Questions

Motivated by the observation that evaluation entropy varies among benchmarks, we propose to further examine the intermediate thought coverage on questions with different levels of difficulty in MATH. According to Fig. 6, the performance gap between EGART Score TopK and the baseline is large on Level 1-4 questions when $\tau = 1$, and the gap narrows as τ decreases due to a large preservation budget. However, in EGART Score Sample, the performance degradation is slight on Level 1-2 questions. This implies the advantage of stochastic exploration on simple tasks, where high-quality candidates are dominant, when the preservation budget is small. However, in the case that weak models (e.g., Qwen2.5-3B) and hard tasks (e.g., Level 4-5) are involved, the performance tends to be unstable and drops significantly. Thus, ranking is a reliable signal given an appropriate preservation budget, whereas sampling is more robust to aggressive budget reduction on easy tasks, especially when LLMs are weak. More discussions regarding the voting case are available in Appendix E.1.

5 Conclusion

This paper starts with the inferior performance of ToT to SC-CoT on mathematical reasoning tasks and focuses on the unreliable self-evaluation of ToT. We conduct both theoretical analyses and empirical studies on both paradigms and find that intermediate thought coverage and high-level information derived from self-evaluation are meaningful in ToT reasoning. Based on these observations, further studies are conducted to validate the efficacy of self-evaluation and ranking. Overall, our work sheds light on the underlying mechanism of ToT and provides meaningful directions for improvements.

AI USE STATEMENT

In this paper, we mainly use AI tools to check hallucination references and polish the manuscript. We DO NOT use AI tools to write the paper and conduct experiments. The authors take full responsibility for the final manuscript, including its claims, citations, contents, and experimental results.

ETHICS STATEMENT

Our work does not contain ethics concerns.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility, in this work, we have clearly provided the experimental settings, including the type of LLMs, deployment details, hyperparameter settings, benchmarks, random seeds, and other concrete experimental details in both the main part and the appendices of this paper. Moreover, this paper also includes theoretical results. To make sure that all results can be reproduced, we provided the concrete proofs regarding these results in the appendices.

REFERENCES

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

Pranjal Aggarwal, Aman Madaan, Yiming Yang, et al. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 2003.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)*, 2015.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Furui Cheng, Vilém Zouhar, Simran Arora, Mrinmaya Sachan, Hendrik Strobelt, and Mennatallah El-Assady. Relic: Investigating large language model responses using self-consistency. In *Proceedings of the 2024 CHI conference on human factors in computing systems*, 2024.

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

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics (ACL Findings)*, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, 2024.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, et al. Efficiently scaling llm reasoning with certaindex. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *International Conference on Learning Representations*, volume 2026, pp. 94355–94377, 2026.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. Rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 1963.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031*, 2025.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*, 2023.

Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Ziyi Zhu, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, et al. Rewarding chatbots for real-world engagement with millions of users. *arXiv preprint arXiv:2303.06135*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

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

Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. Mugglemath: Assessing the impact of query and response augmentation on math reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024a.

Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. Chain of code: Reasoning with a language model-augmented code emulator. In *International Conference on Machine Learning (ICML)*, 2024b.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *International Conference on Learning Representations (ICLR)*, 2024c.

Zenan Li, Zhi Zhou, Yuan Yao, Yu-Feng Li, Chun Cao, Fan Yang, Xian Zhang, and Xiaoxing Ma. Neuro-symbolic data generation for math reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024d.

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

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, et al. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *International Conference on Learning Representations*, volume 2025, pp. 49573–49609, 2025.

Bhuvanashree Murugadoss, Christian Poelitz, Ian Drosos, Vu Le, Nick McKenna, Carina Suzana Negreanu, Chris Parnin, and Advait Sarkar. Evaluating the evaluator: Measuring llms' adherence to task evaluation instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Sungjin Park, Xiao Liu, Yeyun Gong, and Edward Choi. Ensembling large language models with process reward-guided tree search for better complex reasoning. 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)*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Xiangyu Zhu, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. In *International Conference on Learning Representations (ICLR)*, 2026.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. In *Advances in neural information processing systems (NeurIPS)*, 2023.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in neural information processing systems (NeurIPS)*, 2022b.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations (ICLR)*, 2024.

Mingfeng Xue, Dayiheng Liu, Wenqiang Lei, Xingzhang Ren, Baosong Yang, Jun Xie, Yidan Zhang, Dezhong Peng, and Jiancheng Lv. Dynamic voting for efficient reasoning in large language models. In *Findings of the Association for Computational Linguistics (EMNLP)*, pp. 3085–3104, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Ruixin Yang, Dheeraj Rajagopal, Shirley Anugrah Hayati, Bin Hu, and Dongyeop Kang. Confidence calibration and rationalization for llms via multi-agent deliberation. *arXiv preprint arXiv:2404.09127*, 2024b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in neural information processing systems (NeurIPS)*, 2023.

Yuxuan Yao, Han Wu, Zhijiang Guo, Biyan Zhou, Jiahui Gao, Sichun Luo, Hanxu Hou, Xiaojin Fu, and Linqi Song. Learning from correctness without prompting makes llm efficient reasoner. In *Conference on Language Modeling*, 2024.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, et al. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*, 2024.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. Fact-and-reflection (far) improves confidence calibration of large language models. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2024.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for llm reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Detailed Related Work

Stepwise Reasoning Paradigm. With the development of large language models, reasoning (Wei et al., 2022a), as an important emergent capability, has attracted more and more attention from the community. As a classic pioneering work, Chain-of-Thought (Wei et al., 2022b) significantly enhances the reasoning capability of LLMs by having LLMs perform reasoning in a stepwise way. Specifically, by plugging "Let's think step by step" into the prompt, LLMs will break the problem into a series of sub-problems and generate the answer logically. Similarly, Zhou et al. (2023) proposed Least-to-Most to break the challenging problems into a sequence of easy sub-problems and address the problems by sequentially solving those sub-problems. However, since CoT performs the reasoning in a black-box way, an erroneous intermediate thought may misguide the reasoning direction, thereby leading to unstable and poor performance. As a solution, test-time scaling reasoning paradigms (Snell et al., 2024; Welleck et al., 2024; Brown et al., 2024; Irvine et al., 2023) are proposed to alleviate the effect of potentially erroneous intermediate thoughts with additional computations. Specifically, self-consistency CoT (SC-CoT) (Wang et al., 2023) is proposed to sample multiple CoT reasoning trajectories for every single query and perform voting among all candidates for the final results. In SC-CoT, the erroneous intermediate thoughts are assumed to be an occasional case and can be alleviated by selecting the most frequent answer. Meanwhile, different from SC-CoT, Tree-of-Thoughts (ToT) (Yao et al., 2023) proposes to explicitly perform reasoning in a tree-search manner. Specifically, ToT has LLMs generate multiple intermediate thoughts explicitly in each step and evaluate these candidate thoughts to determine the reasoning trajectories. Compared to SC-CoT, which enhances the robustness of reasoning by exploring the consistency among diverse trajectories, ToT focuses on finding the optimal reasoning path through stepwise verifiable evaluations. Although both SC-CoT help alleviate the uncertainty and potential mistakes in reasoning and, in turn, improve the exploration capability of LLMs in reasoning, due to its test-time scaling property, more tokens are consumed as a cost for the impressive performance. So far, many improvements have been proposed to solve this problem. Specifically, ESC (Li et al., 2024c) proposes to set an early-stopping rule for SC-CoT to reduce the reasoning cost. Specifically, ESC splits the entire sampling size into batches. Once the answers in a batch are all the same, the sampling will stop. Moreover, Adaptive-Consistency (Aggarwal et al., 2023) notices that the gap between the answers with the most votes and the second most votes depicts the certainty of the LLMs to the given query, thereby proposing to perform an early stop when an answer takes a dominant position in the candidate set. Dynamic Voting (Xue et al., 2023) further formulates the SC-CoT into a multi-round, incremental voting process. Specifically, Dynamic Voting first samples a set of answers. If the consistency reaches a predefined threshold, the reasoning will stop and the answers will be generated. Otherwise, another sampling will be performed in the next round until the consistency reaches the threshold. Dynasor (Fu et al., 2025) finds that existing reasoning methods exhibit answer stabilization, where the intermediate solutions often cease to change after a certain point, thereby proposing to measure the evolving stability and evaluate whether further computation is likely to obtain the correct answers. Accordingly, the thoughts that are less likely to reach the correct final answers will be terminated. Similarly, DeepConf (Fu et al., 2026) is also proposed recently to perform reasoning trajectory selection in the early phase of reasoning. In addition to the aforementioned methods, other reasoning paradigms are also developed to further improve the reasoning performance and efficiency through either search strategies (Guan et al., 2025; Zhang et al., 2024; Park et al., 2025) or RL (Yu et al., 2023; Li et al., 2024a; Ying et al., 2024; Li et al., 2024d; Zhou et al., 2025; Luo et al., 2025).

LLM Confidence Estimation. Due to the stochastic characteristics in generative models, the responses sampled from the LLMs may contain erroneous and spurious information. In order to improve the generation quality, confidence estimation has become an essential aspect in responses evaluation. So far, confidence estimation can be roughly divided into subjective and objective evaluations. For subjective evaluation, the confidence is measured by directly asking LLMs to score the responses (Tian et al., 2023; Xiong et al., 2024; Kadavath et al., 2022), such as multi-agent deliberation (Yang et al., 2024b) and reflection Dhuliawala et al. (2024); Zhao et al. (2024). In objective evaluation, mainstream works leverage the internal states to measure the confidence of LLMs, such as perplexity (i.e., the geometric mean of the log-probabilities of LLM predictions) (Huang et al., 2023; Duan et al., 2024; Blei et al., 2003). Such a metric is usually adopted to evaluate the fluency (Murugadoss et al., 2025) and the quality of prompts (Yao et al., 2024). In recent work (Zhou et al., 2025), the consistency in responses (Wang et al., 2023; Cheng et al., 2024; Huang et al., 2025)/internal states (Zhou et al., 2025) is also treated as a kind of confidence. In this paper, we also explore the confidence of self-evaluation of LLMs through the entropy of voting behaviors. Actually,

our work partially shares the same idea as the Certaindex in Dynasor Fu et al. (2025). In Dynasor, the diversity of intermediate answers is measured through normalized entropy to evaluate the certainty of LLMs and determine whether the reasoning should be terminated. However, in our work, the entropy is leveraged to measure the certainty in self-evaluation of LLMs and determine the preserved intermediate thoughts in the next reasoning step.

B Proofs of Theoretical Results in Section 2.2

In this section, we provide detailed proofs for the theoretical results of the capabilities of the two reasoning paradigms (Theorem 2.2 and Theorem 2.4) discussed in Section 2.2.

B.1 Proofs for SC-CoT (Theorem 2.2)

Theorem 2.2. (**SC-CoT**) *Consider an effective answer set $\mathcal{A}_\theta$ ($|\mathcal{A}_\theta| < \infty$). Define the probability of LLMs generating correct answers as $p_y := q(y)$ and the probability of generating any erroneous answer as $q_a, \forall a \in \mathcal{A}_\theta \backslash \{y\}$, so that $\sum_{a \in \mathcal{A}_\theta \backslash \{y\}} q_a = 1 - p_y$. Define $q_{\max} := \max_{a \in \mathcal{A}_\theta \backslash \{y\}} q_a$. Then, the probability that SC-CoT successfully infers the correct answer is bounded as follows:*

$$P(N_{\text{ans}} > N_{\text{err}}) \begin{cases} \leq \exp\left(-\frac{1}{2} N_{\text{T}} \left(p_y - q_{\max}\right)^2\right), & p_y \leq q_{\max}, \\ \geq 1 - \sum_{a \in \mathcal{A}_\theta \backslash \{y\}} \exp\left(-\frac{1}{2} N_{\text{T}} \left(p_y - q_a\right)^2\right), & p_y > q_{\max}. \end{cases}$$

Proof. Given a fixed $a \in \mathcal{A}_\theta \backslash \{y\}$, we first define $X_i = \mathbb{1}\left(\hat{y}^{(i)} = y\right) - \mathbb{1}\left(\hat{y}^{(i)} = a\right)$. Then, we have a set of *i.i.d.* variables $\{X_i\}_{i=1}^{N_{\text{T}}}$, where $X_i \in \{-1, 0, 1\}$, given the N_{T} reasoning trajectories *i.i.d.* sampled from LLMs and $\mathbb{E}\left(X_i\right) = p_y - q_a$. Thus, we can obtain $S_a = N_{\text{ans}} - U_a = \sum_{i=1}^{N_{\text{T}}} X_i$, where $U_a = \sum_{i=1}^{N_{\text{T}}} \mathbb{1}\left(\hat{y}^{(i)} = a\right)$. Correspondingly, we have $\mathbb{E}[S_a] = N_{\text{T}}\left(p_y - q_a\right)$.

We use Hoeffding's inequality (Hoeffding, 1963) to bound the deviation of S_n from its expectation. Given independent random variables $X_i \in [a_i, b_i]$, and $S_n = \sum_{i=1}^{n} X_i$, Hoeffding's inequality is formulated as:

$$P(S_n - \mathbb{E}[S_n] \geq t) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$
$$P(S_n - \mathbb{E}[S_n] \leq -t) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

By Hoeffding's inequality, if $p_y > q_a$, we consider $S_a \leq 0$. Since $a_i = -1$ and $b_i = 1$, $\forall i \in \{1, ..., N_{\text{T}}\}$, we have

$$\begin{aligned} P\left(S_a - \mathbb{E}[S_a] \leq -N_{\text{T}}\left(p_y - q_a\right)\right) &\leq \exp\left(-\frac{2N_{\text{T}}^2\left(p_y - q_a\right)^2}{4N_{\text{T}}}\right) \\ &= \exp\left(-\frac{1}{2} N_{\text{T}}\left(p_y - q_a\right)^2\right). \end{aligned}$$

Then, we extend the conclusion to a global case where the count of the correct answer is larger than the count of the most frequent erroneous answer (i.e., $P(N_{\text{ans}} > N_{\text{err}})$). Correspondingly, we here consider $p_y > q_{\max}$ ($q_{\max} = \max_{a \in \mathcal{A}_\theta \backslash \{y\}} q_a$). Thus, we further have

$$\begin{aligned} P\left(N_{\text{ans}} \leq N_{\text{err}}\right) &= P\left(N_{\text{ans}} \leq \max_{a \in \mathcal{A}_\theta \backslash \{y\}} U_a\right) \\ &= P\left(\bigcup_{a \in \mathcal{A}_\theta \backslash \{y\}} \{N_{\text{ans}} \leq U_a\}\right) \\ &\leq \sum_{a \in \mathcal{A}_\theta \backslash \{y\}} P\left(N_{\text{ans}} \leq U_a\right) \\ &\leq \sum_{a \in \mathcal{A}_\theta \backslash \{y\}} \exp\left(-\frac{1}{2} N_{\text{T}}\left(p_y - q_a\right)^2\right) \end{aligned}$$

Correspondingly, we have

$$P\left(N_{\text{ans}} > N_{\text{err}}\right) \geq 1 - \sum_{a \in \mathcal{A}_\theta \setminus \{y\}} \exp\left(-\frac{1}{2} N_{\text{T}} \left(p_y - q_a\right)^2\right).$$

Otherwise, if $p_y \leq q_{\max}$, we consider $S_{a^*} > 0$, where $a^* := \arg\max_{a \in \mathcal{A}_\theta \setminus \{y\}} q_a$. Then, we have $N_{\text{ans}} > N_{\text{err}} \implies N_{\text{ans}} > U_{a^*}$. Thus, we have

$$P\left(N_{\text{ans}} > N_{\text{err}}\right) \leq P(N_{\text{ans}} > U_{a^*}) \leq \exp\left(-\frac{1}{2} N_{\text{T}} \left(p_y - q_{\max}\right)^2\right).$$

□

B.2 Proofs for Theorem 2.4

Theorem 2.4. (**Single-step ToT**) *Given a set of intermediate thoughts $\mathcal{Z}_i$ of the i-th step in ToT reasoning, where at least one correct intermediate thought is included. N_{v} evaluations are independently conducted for each intermediate thought $z \in \mathcal{Z}_i$ and a score $v_z^{(k)} \in [0,1]$ is obtained at the k-th evaluation. Let $\phi_{i,z^+} = \mathbb{E}[v_{z^+}^{(k)}]$ for $z^+ \in \mathcal{C}_i^+$ and $\rho_{i,z^-} = \mathbb{E}[v_{z^-}^{(k)}]$ for $z^- \in \mathcal{C}_i^-$ denote the expected scores that a correct and an incorrect intermediate thought is selected. Let $\phi_i = \max_{z \in \mathcal{C}_i^+} \phi_{i,z}$ obtained at z_i^+ and $\Delta_i = \phi_i - \rho_i^{(N_{\text{p}})}$ denote the margin between the best correct intermediate thought and the N_{p}-th most favored incorrect intermediate thought. With the mild assumptions in Assumption 2.3, the probability of ToT successfully selecting a correct candidate thought is bounded*

$$P\left(E_i\right) \begin{cases} \leq \min\left\{1, |\mathcal{Z}_i| N_{\text{p}} \exp\left(-\frac{1}{2} N_{\text{v}} \Delta_i^2\right)\right\}, & \Delta_i < 0, \\ \geq 1 - \left(|\mathcal{Z}_i| - 1\right) \exp\left(-\frac{1}{2} N_{\text{v}} \Delta_i^2\right), & \Delta_i > 0, \end{cases}$$

where E_i denotes the event that at least one selected intermediate thought is correct given the set $\mathcal{Z}_i$.

Proof. According to assumptions, at the i-th step, given a candidate pool $\mathcal{Z}_i$, N_{v} evaluations are conducted independently for all candidates. For any fixed correct intermediate thought $z^+ \in \mathcal{C}_i^+$ and incorrect intermediate thought $z^- \in \mathcal{C}_i^-$, in the k-th evaluation, let $X_k = v_{z^+}^{(k)} - v_{z^-}^{(k)} \in [-1, 1]$. Then, we further have $S = N_{i,z^+} - N_{i,z^-} = \sum_{k=1}^{N_{\text{v}}} X_k$, where N_{i,z^+} and N_{i,z^-} denote the aggregated evaluation of $z^+ \in \mathcal{C}_i^+$ and $z^- \in \mathcal{C}_i^-$, respectively, over all evaluations, $X_1, X_2, \ldots, X_{N_{\text{v}}}$ are *i.i.d.* given $\mathcal{Z}_i$. Thus, we then have $\mathbb{E}[S] = N_{\text{v}}\left(\phi_{i,z^+} - \rho_{i,z^-}\right)$.

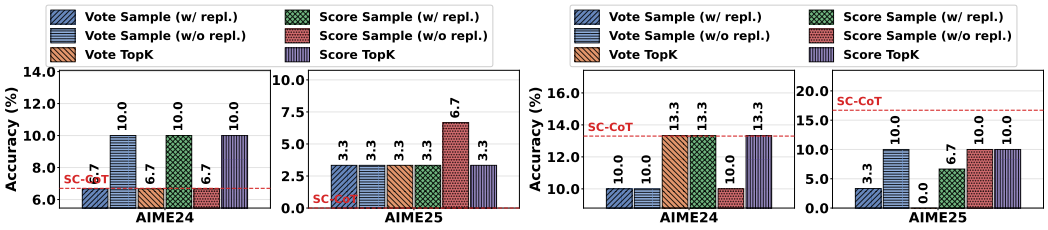
With Hoeffding's Inequality, for any fixed correct intermediate thought $z^+ \in \mathcal{C}_i^+$ and incorrect intermediate thought $z^- \in \mathcal{C}_i^-$, if $\phi_{i,z^+} > \rho_{i,z^-}$, we consider $S \leq 0$. Then, we have

$$\begin{aligned} P\left(S - \mathbb{E}\left[S\right] \leq -N_{\text{v}}\left(\phi_{i,z^+} - \rho_{i,z^-}\right)\right) &\leq \exp\left(-\frac{2N_{\text{v}}^2\left(\phi_{i,z^+} - \rho_{i,z^-}\right)^2}{4N_{\text{v}}}\right) \\ &= \exp\left(-\frac{1}{2} N_{\text{v}}\left(\phi_{i,z^+} - \rho_{i,z^-}\right)^2\right). \end{aligned}$$

Moreover, if $\phi_{i,z^+} < \rho_{i,z^-}$, we consider $S \geq 0$. Then, we have

$$P\left(S - \mathbb{E}\left[S\right] \geq N_{\text{v}}\left(\rho_{i,z^-} - \phi_{i,z^+}\right)\right) \leq \exp\left(-\frac{1}{2} N_{\text{v}}\left(\phi_{i,z^+} - \rho_{i,z^-}\right)^2\right).$$

Then, we extend the conclusion to the global case. We first consider $\Delta_i > 0$. If $|\mathcal{C}_i^-| < N_{\text{p}}$, the selected set necessarily contains a correct candidate intermediate thought and $P(E_i^{\text{no}}) = 0$, where E_i^{no} denotes the event that none of the correct intermediate thoughts are selected. Otherwise, all N_{p} selected candidates are incorrect, so z_i^+ is not selected, and each selected z^- satisfies $N_{i,z^-} \geq N_{i,z_i^+}$. Thus, $E_i^{\text{no}} \subseteq \bigcup_{z^- \in \mathcal{C}_i^- : \rho_{i,z^-} \leq \rho_i^{(N_{\text{p}})}} \left\{N_{i,z_i^+} \leq N_{i,z^-}\right\}$. Since at most $N_{\text{p}} - 1$ incorrect intermediate



(a) Acc. of Different Strategies with Qwen2.5-3B (b) Acc. of Different Strategies with Qwen2.5-7B

Figure 7: **Empirical Study on Different Evaluation and Selection Strategies in ToT (on AIME24 and AIME25).** Consistent with the cases on GSM8K and MATH, on AIME benchmarks, on the one hand, ToT equipped with sampling without replacement tends to achieve better performance than ToT equipped with sampling with replacement. Meanwhile, scoring contributes to improving the performance of the variant with sampling with replacement. These consistent phenomena demonstrate the significance of the coverage of intermediate thoughts. On the other hand, the advantage of top-k remains and better performance is achieved than ToT with sampling w/o replacement in most cases.

thoughts have $\rho_{i,z^-} > \rho_i^{(N_\mathrm{p})}$, where $\rho_i^{(N_\mathrm{p})}$ denotes the N_p-th largest value, at least one selected $z^- \in \mathcal{C}_i^-$ satisfies $\rho_{i,z^-} \leq \rho_i^{(N_\mathrm{p})}$. Thus, we have

$$
\begin{aligned}
P\left(E_i^{\mathrm{no}}\right) &\leq P\left(\bigcup_{z^- \in \mathcal{C}_i^- : \rho_{i,z^-} \leq \rho_i^{(N_\mathrm{p})}} \left\{N_{i,z_i^+} \leq N_{i,z^-}\right\}\right) \\
&\leq \sum_{z^- \in \mathcal{C}_i^- : \rho_{i,z^-} \leq \rho_i^{(N_\mathrm{p})}} \exp\left(-\frac{1}{2} N_\mathrm{v} \left(\phi_i - \rho_{i,z^-}\right)^2\right) \\
&\leq \left(|\mathcal{Z}_i| - 1\right) \exp\left(-\frac{1}{2} N_\mathrm{v} \Delta_i^2\right)
\end{aligned}
$$

Correspondingly, for the event E_i that at least one selected intermediate thought is correct, the probability is bounded as

$$
P\left(E_i\right) \geq 1 - \left(|\mathcal{Z}_i| - 1\right) \exp\left(-\frac{1}{2} N_\mathrm{v} \Delta_i^2\right).
$$

Next, we consider $\Delta_i < 0$, which requires $|\mathcal{C}_i^-| \geq N_\mathrm{p}$ for $\rho_i^{(N_\mathrm{p})}$ to be well-defined. Let $\mathcal{T}_i \subseteq \mathcal{C}_i^-$ denote the set of top-N_p intermediate thoughts in the set $\mathcal{C}_i^-$. If any $z^+ \in \mathcal{C}_i^+$ is selected, at least one $z^- \in \mathcal{T}_i$ is not selected, and it satisfies $N_{i,z^+} \geq N_{i,z^-}$. Thus, we have $E_i \subseteq \bigcup_{z^+ \in \mathcal{C}_i^+} \bigcup_{z^- \in \mathcal{T}_i} \left\{N_{i,z^+ \geq N_{i,z^-}}\right\}$. Since $\phi_{i,z^+} - \rho_{i,z^-} \leq \phi_i - \rho_i^{(N_\mathrm{p})} = \Delta_i < 0$, we then have

$$
\begin{aligned}
P\left(E_i\right) &\leq \sum_{z^+ \in \mathcal{C}_i^+} \sum_{z^- \in \mathcal{T}_i} P\left(N_{i,z^+} \geq N_{i,z^-}\right) \\
&\leq |\mathcal{Z}_i| N_\mathrm{p} \exp\left(-\frac{1}{2} N_\mathrm{v} \Delta_i^2\right)
\end{aligned}
$$

.

□

C More Details of Empirical Validations in Section 3

In this section, we provide detailed experimental settings and discussions about our empirical studies in Section 3. Specifically, we first provide detailed settings adopted in our empirical studies. Then, we further provide more discussions and explanations regarding our experimental results.

C.1 General Experimental Settings

We first introduce experimental details, including large language models and implementation details.

C.1.1 LARGE LANGUAGE MODELS

In our empirical motivation studies, we mainly consider two Qwen2.5-Instruct models (Yang et al., 2024a): Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, which are widely adopted in previous LLM reasoning work due to their stable performance. Here, we adopt these two models to evaluate the reasoning paradigms under different scales of LLMs. In addition, since the two models were released earlier than some benchmarks (e.g., AIME25), we can ensure that their prior knowledge does not cover these benchmarks. Thus, we can evaluate the two reasoning paradigms in an objective way.

C.1.2 DETAILS OF STUDY ON DIVERSITY OF ANSWERS

In the study on the reasoning performance of SC-CoT, we evaluate the performance of SC-CoT on all testing data of GSM8K (Cobbe et al., 2021) and AIME24 (Zhang & Math-AI, 2024), respectively, with different LLMs (i.e., Qwen2.5-3B/7B) and sampling sizes (i.e., 40 and 80). Specifically, for each query, we examine the uniqueness of answers (i.e., the number of unique answers among all sampled reasoning trajectories) and whether the correct final answer is obtained through voting. To avoid noisy results, we dropped the cases that contain fewer than 20 samples in the GSM8K benchmark.

C.1.3 IMPLEMENTATION OF TREE-OF-THOUGHTS REASONING PARADIGM

In our analyses on ToT, we mainly follow the implementation details in the original paper of ToT (Yao et al., 2023). Specifically, in our experiments, we set the breadth and the depth of the reasoning tree, respectively, as 5 and 3. For each node in an intermediate reasoning step, we have LLMs generate 5 possible next steps. Then, we request LLMs to evaluate these candidates. The evaluation is performed 10 times. Then, we normalize the evaluation results to obtain the probability/rank for each candidate thought. Further, we ask LLMs to sample 50% of all candidates as the branches (with a ceiling operation) for the next step. For example, in our default settings with breadth as 5 and depth as 3, given a query, the number of branches after each of the 3 reasoning steps is 3, 9, and 27, respectively.

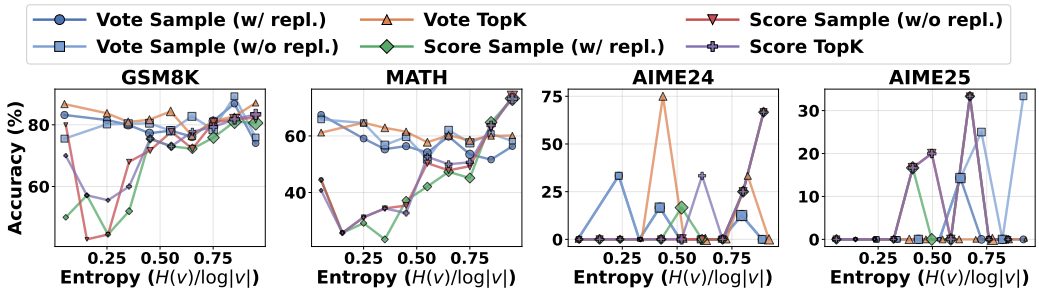
C.1.4 EVALUATION AND SELECTION STRATEGIES ADOPTED IN TOT

A key component of ToT is the self-evaluation and thought selection step between two intermediate reasoning steps. Typically, all candidate intermediate thoughts will be evaluated by LLMs and then a portion of them will be selected for the next step. In the conventional ToT (Yao et al., 2023), both self-evaluation and thought selection have two options. Specifically, on the one hand, self-evaluation can be conducted by either voting or scoring. In voting, all candidates will be fed into LLMs, and LLMs are required to select the optimal one from the candidates. In scoring, all candidate thoughts will be fed into LLMs one by one to get an evaluation in "sure", "impossible", and "likely". Then, a score will be assigned to the thought based on the feedback. On the other hand, thought selection is performed by either sampling or greedy selection. Specifically, sampling means sampling a portion of thoughts from the candidates according to the probabilities obtained from the evaluation. In sampling, we also have "sampling with replacement" and "sampling without replacement". Moreover, greedy selection means selecting the top-k thoughts from the candidates according to the rank of the scores.

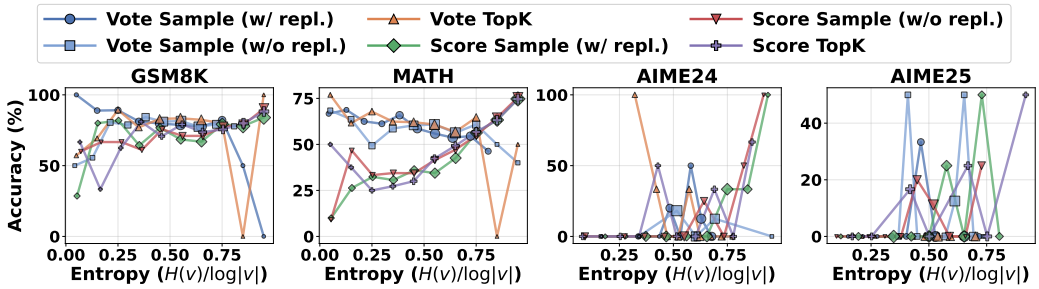
C.2 PROBE BEHAVIORS OF VOTING AND SCORING IN REASONING

In Section 3.2, we investigate how different self-evaluation and intermediate thought selection strategies affect the performance of ToT. An important observation is that self-evaluation in ToT is not reliable. Specifically, when the voting signal is adopted for intermediate thought selection, the selected candidate, which wins among all intermediate thoughts, fails to develop the correct reasoning answer. However, adopting scoring for evaluation facilitates improving the reasoning performance of ToT (cf. Figs. 3 & 7). Intuitively, due to the differences between strategies, where voting aims to select an optimal intermediate thought while scoring aims at evaluating the quality of the candidate pool, the evaluation distribution derived from voting tends to be sharp and the selected option may be biased. In addition to this intuition, in this section, we propose to further conduct empirical studies on intermediate evaluations to investigate the behaviors of the two self-evaluation strategies in reasoning.

The results are visualized in Figs. 12 $\sim$ 19. According to the reported results, we can observe that there exists a position bias when voting is adopted as the self-evaluation strategy. Specifically, when equipped with voting, both LLMs prefer the intermediate thoughts that are ranked in the first several positions. However, when adopting scoring as the self-evaluation strategy, we can hardly observe such a phenomenon in evaluations. With this observation, we conjecture that the selected intermediate thoughts in ToT with voting are mainly determined by the sequences after random shuffling. For example, a set of intermediate thoughts of low quality yet at the top of the randomly



(a) Entropy of the First Evaluation Round on All Benchmarks with Qwen2.5-7B



(b) Entropy of the Second Evaluation Round on All Benchmarks with Qwen2.5-7B

Figure 8: **Empirical Study on the Correlation between the Performance and the Entropy of Evaluation Behaviors of ToT with Qwen2.5-7B.** We here examine the correlation between the reasoning performance and the entropy of evaluation behaviors (e.g., the normalized entropy of the voting/scoring vector: $H(\boldsymbol{v})/\log(|\boldsymbol{v}|)$) in both evaluation rounds of ToT on all benchmarks with different evaluation strategies. The phenomena are consistent with the phenomena observed in Fig. 4

shuffled sequence may be selected for the next reasoning step, thereby resulting in poor performance. This phenomenon can further help explain why ToT equipped with sampling with replacement fails to achieve superior performance to ToT equipped with sampling without replacement. Specifically, the high-ranked incorrect intermediate thoughts are voted in evaluations. Then, when applying sampling with replacement for thought selection, the incorrect intermediate thought will be repeatedly sampled due to the concentrated voting, which, in turn, results in poor reasoning performance.

D Implementation Details of Experiments

In this section, we provide detailed implementation settings adopted in the experiments in Section 4. Specifically, we provide comprehensive details about large language models, datasets, and prompts.

D.1 Large Language Models

All our experiments in this paper are conducted on Qwen series LLMs (Yang et al., 2024a), including Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. In practical implementations, we set the temperature coefficient to 1.0, the top-p parameter to 1.0, and the maximum length of tokens to 32768. The large language models are deployed on a server with two NVIDIA RTX 3090 GPUs to perform reasoning.

D.2 Baselines

In our experiments, the following reasoning paradigms are adopted as baselines: Chain-of-Thought (CoT) (Wei et al., 2022b), Self-Consistency Chain-of-Thought (SC-CoT) (Wang et al., 2023), and Tree-of-Thoughts (ToT) (Yao et al., 2023). The first two are sampling-based paradigms, while the latter is a search-based paradigm. For the implementation of SC-CoT, we randomly sample 40 reasoning paths with a temperature of 1.0. The implementation of ToT is available in Appendix C.1.3.

D.3 Benchmarks

In this paper, we focus on studying the performance of two different reasoning paradigms, SC-CoT and ToT, on mathematical reasoning tasks. Thus, we mainly consider four mainstream mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), AIME 24 (Zhang & Math-AI, 2024), and AIME 25 (Zhang & Math-AI, 2025), ranging from easy to hard reasoning, in our experiments. More details about these benchmarks are provided as follows:

- **GSM8K.** GSM8K, also known as Grade School Math 8K, is a popular mathematical benchmark to evaluate the mathematical reasoning capability of large language models. In this benchmark, 8,500 high-quality linguistically diverse grade school math word problems are contained. Typically, all problems in GSM8K require solutions with 2-8 steps involving basic arithmetical operations.
- **MATH.** MATH is a dataset that contains 12,500 competition math problems. The MATH benchmark can be used to evaluate the mathematical reasoning capabilities of LLMs under more challenging scenarios. Specifically, the questions in MATH refer to various kinds of mathematical problems, including algebra, geometry, number theory, and combinatorics. In this paper, for the time concern, we randomly select 2,000 samples from the benchmark for evaluation, with the random seed of 42.
- **AIME 24 & 25.** AIME, also known as American Invitational Mathematics Examination, consists of 30 challenging Olympiad-level mathematical problems from AIME I and AIME II competitions. Similar to MATH, the problems in AIME also refer to various kinds of fields, ranging from algebra to combinatorics. Compared to GSM8K and MATH, AIME is the most difficult benchmark.

D.4 PROMPTS

In this section, we provide the details about prompts adopted in our experiments, including the prompts used in Chain-of-Thought reasoning and Tree-of-Thoughts reasoning. To avoid as much effect of prompts as possible, we write prompts as concisely as possible with only necessary instructions.

CoT Prompt (for CoT and SC-CoT)

`<QUESTION>`
Please solve the problem step by step.

Response format (MUST be strictly followed. DO NOT include any other formats except for the given XML format):
<think>YOUR THINKING HERE</think>
<answer>YOUR FINAL ANSWER ONLY, NO OTHER TEXT</answer>.

ToT Propose Prompt for Initial Step

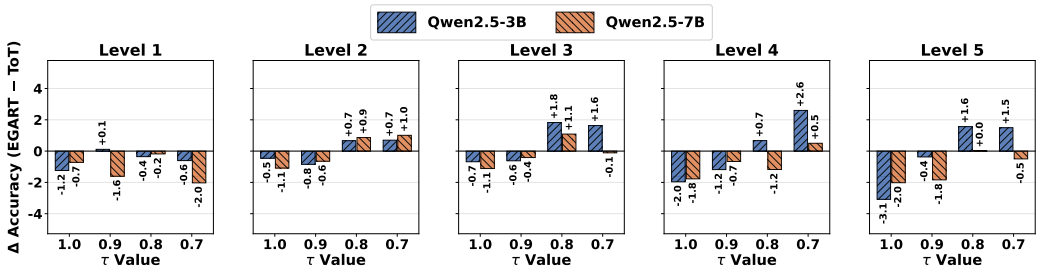
Instruction: You need to solve the given query step by step. Here, you need to provide a possible next step (with corresponding calculation processes) based on your thinking regarding the given query with the given XML format:
<think>YOUR THINKING HERE</think>
<intermediate>YOUR POSSIBLE NEXT STEP HERE</intermediate>
DO NOT include any other formats except for the given XML format.

Input: `<Question>` Please propose a possible next step:

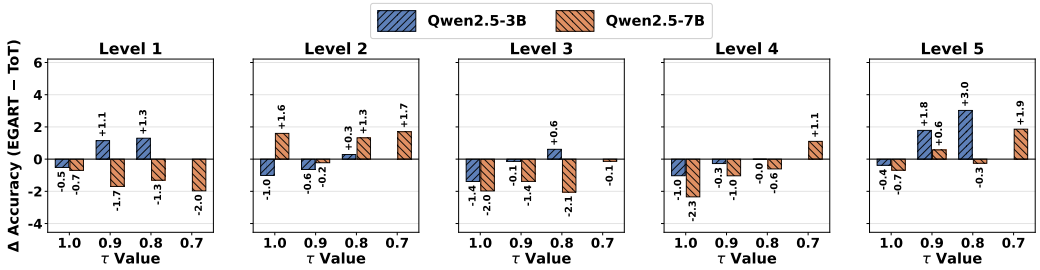
ToT Propose Prompt for Intermediate Step

Instruction: You need to solve the given query step by step. Here, you need to provide a possible next step (with corresponding calculation processes) based on your thinking regarding the given query and the previous step with the given XML format:
<think>YOUR THINKING HERE</think>
<intermediate>YOUR POSSIBLE NEXT STEP HERE</intermediate>
DO NOT include any other formats except for the given XML format.
You should summarize the previous solution step and provide a possible next step in the <intermediate> tag.

Input: `Question`
The previous step: `Previous Step`
Please propose a possible next step:



(a) Differences between EGART Vote TopK and ToT baseline with varied τ values



(b) Differences between EGART Vote Sampling and ToT baseline with varied τ value

Figure 9: **An Investigation on Performance Changes of EGART (with voting) with τ Values on Different Levels of Questions**. We here report the performance changes of EGART Vote TopK and EGART Vote Sample, respectively, with various τ on different levels of questions in MATH dataset.

ToT Propose Prompt for Final Step

Instruction: You need to solve the given query step by step. Here, you need to return the final answer (with corresponding calculation processes) based on your thinking regarding the given query and the previous step with the given XML format:
<think>YOUR THINKING HERE</think>
<answer>FINAL ANSWER, NUMERICAL VALUE ONLY</answer>
DO NOT include any other formats except for the given XML format.

Input: `Question`
The previous step: `Previous Step`
Please return the final answer:

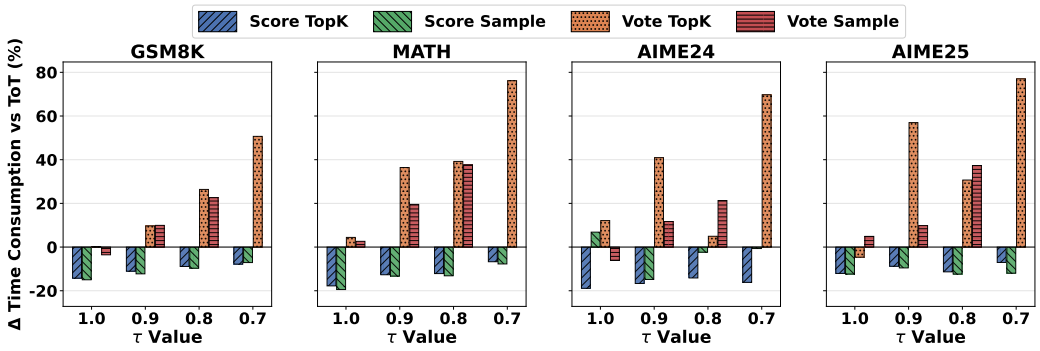
E More Experimental Results and Discussions

In this section, we provide more detailed discussions regarding the experiments in the main part. Moreover, we also further compare the efficiency and resource consumptions of EGART and ToT.

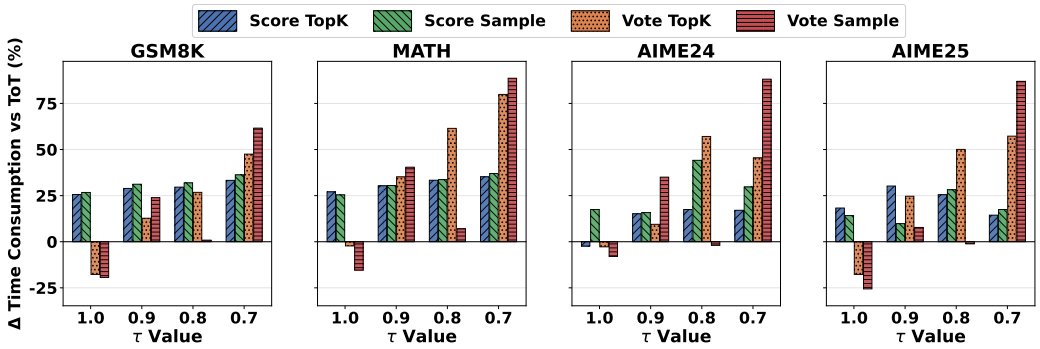
E.1 More Qualitative Results of Ranking

In Section 4.3, we provide qualitative analyses on the ranking information (i.e., top-k) derived from self-evaluation. In summary, when adopting score-based ranking for intermediate thought selection, given an appropriate preservation budget, good performance can be achieved. In contrast, when the preservation budget is aggressively reduced, sampling usually achieves better performance via stochastic exploration. In this part, we further discuss the qualitative results of EGART under voting.

The results are visualized in Fig. 9. In most cases, with τ decreases, the performance of EGART tends to be improved when top-k is adopted as the selection strategy. This further highlights the importance of the preservation budget when exploiting ranking information for intermediate thoughts. However, inconsistent with the case in scoring, even on simple tasks with an aggressive reduction of the preservation budget ($\tau = 1.0$), the performance gap between EGART Vote TopK and EGART Vote Sample is not obvious. Meanwhile, different from the scoring case, the performance of EGART Vote Sample is not stable. For example, as τ decreases, the performance on simple tasks (e.g., Level 1 questions) drops significantly, especially on Qwen2.5-7B. Moreover, on other levels of questions, we can also observe such unstable phenomena. The phenomenon regarding top-k and sampling under voting further demonstrates the reliability of ranking, consistent with the observation in Section 4.2



(a) Time Consumption Analyses of EGART with Different Evaluation and Selection Strategies on Qwen2.5-3B



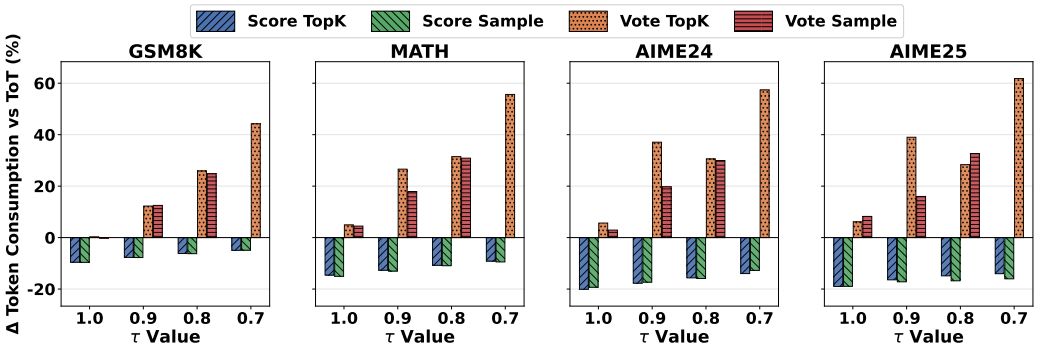
(b) Time Consumption Analyses of EGART with Different Evaluation and Selection Strategies on Qwen2.5-7B

Figure 10: **Time Consumption Analyses of Various Evaluation and Selection Strategies.**

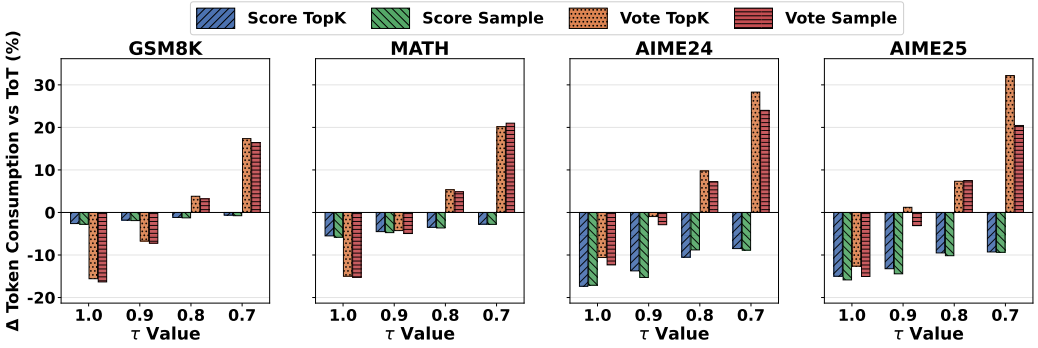
E.2 Efficiency and Resource Analyses on EGART

In this work, we propose to probe the properties of self-evaluation and intermediate thought selection via an intervention tool, entropy-guided adaptive reasoning tree (EGART), to adaptively control the preservation budget of intermediate thoughts. In this section, we further analyze the time and computational resource consumption (i.e., time and token consumption) for efficiency analysis.

Time and computational consumption are visualized in Figs. 10 & 11. For time efficiency, we can observe that EGART with voting tends to consume more time in reasoning than the ToT baseline, while EGART with scoring consumes less time in reasoning than the ToT baseline. Moreover, compared to small models (e.g., Qwen2.5-3B), the EGART with scoring cases consume more time than baselines on large models (e.g., Qwen2.5-7B). For token consumption, we can observe that EGART with scoring tends to consume less tokens than the corresponding baselines, while EGART with voting usually consumes more tokens. Along with the quantitative results, EGART with scoring reveals significant potential in achieving good performance with less resource consumption.



(a) Token Consumption of EGART with Different Evaluation and Selection Strategies on Qwen2.5-3B

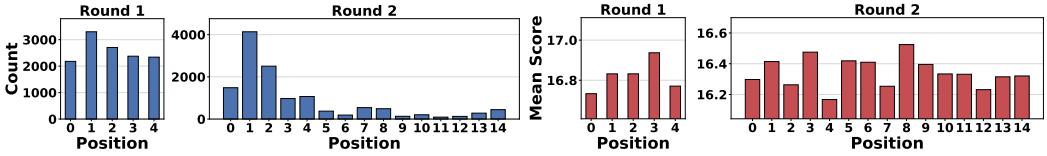


(b) Token Consumption of EGART with Different Evaluation and Selection Strategies on Qwen2.5-7B

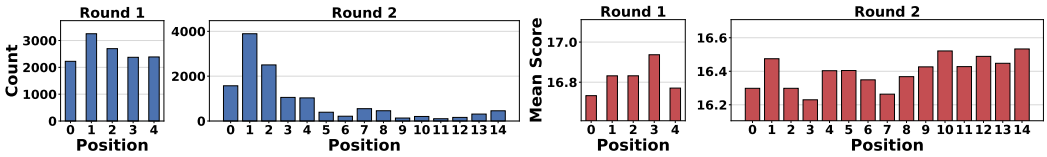
Figure 11: **Token Consumption Analyses of Various Evaluation and Selection Strategies.**

Table 2: Empirical results of accuracy (Acc.) and the *average* number of reasoning paths (S_{avg}) per query in the final reasoning step on four mathematical reasoning benchmarks, GSM8K, MATH, and AIME 24 & 25, with Qwen2.5-Instruct Models. We highlight the best performance in **bold** and the second best performance with an underline. All reported results are the average of seeds 41 and 42.

Methods	GSM8K		MATH		AIME 24		AIME 25	
	Acc. (%) ↑	S_{avg} ↓	Acc. (%) ↑	S_{avg} ↓	Acc. (%) ↑	S_{avg} ↓	Acc. (%) ↑	S_{avg} ↓
`Qwen2.5-3B-Instruct`								
CoT	64.1±2.5	1	41.7±1.8	1	5.0±2.4	1	1.7±2.3	1
CoT-SC	**90.8±0.3**	40	**66.7±0.2**	40	8.4±2.3	40	0.0±0.0	40
ToT Score TopK	89.1±0.2	45	59.5±0.1	45	8.4±2.3	45	5.0±2.4	45
EGART Score TopK ($\tau = 1.0$)	87.2±0.2	32.2	58.0±0.7	26.2	6.7±0.0	20.2	1.7±2.3	18.8
EGART Score TopK ($\tau = 0.9$)	87.7±0.1	35.4	58.2±0.6	28.9	6.7±4.7	21.5	5.0±2.4	21.4
EGART Score TopK ($\tau = 0.8$)	87.6±0.1	37.8	58.2±0.4	31.8	8.4±2.3	23.4	5.0±2.4	25.7
EGART Score TopK ($\tau = 0.7$)	88.1±0.1	39.7	59.2±0.0	34.4	5.0±2.4	26.9	**10.0±4.7**	25.9
ToT Score Sample	88.8±0.3	45	59.7±1.0	45	5.0±2.4	45	5.0±2.4	45
EGART Score Sample ($\tau = 1.0$)	87.0±0.8	32.4	57.8±0.2	26.1	5.0±2.4	20.2	3.3±0.0	21.2
EGART Score Sample ($\tau = 0.9$)	88.0±0.6	35.3	58.4±1.8	29.1	6.7±4.7	20.8	5.0±2.4	21.8
EGART Score Sample ($\tau = 0.8$)	88.0±0.5	37.7	58.7±0.9	32.0	**10.0±4.7**	21.8	5.0±7.1	23.6
EGART Score Sample ($\tau = 0.7$)	88.1±0.7	39.9	59.2±0.4	34.7	8.3±7.1	26.8	**10.0±0.0**	26.1
ToT Vote TopK	88.7±0.1	45	60.1±1.5	45	8.4±2.3	45	1.7±2.3	45
EGART Vote TopK ($\tau = 1.0$)	88.1±0.1	32.6	58.2±1.1	35.1	8.4±2.3	35.8	1.7±2.3	35.8
EGART Vote TopK ($\tau = 0.9$)	88.5±0.2	41.7	59.4±0.2	51.8	6.7±0.0	65.3	5.0±2.4	70.0
EGART Vote TopK ($\tau = 0.8$)	89.2±0.4	55.1	61.4±0.4	58.9	6.7±4.7	61.8	3.3±0.0	58.3
ToT Vote Sample	88.6±0.3	45	59.2±1.4	45	6.7±0.0	45	0.0±0.0	45
EGART Vote Sample ($\tau = 1.0$)	87.6±0.3	32.5	58.8±0.8	35.1	3.3±0.0	35.5	3.3±0.0	36.9
EGART Vote Sample ($\tau = 0.9$)	88.9±0.1	41.9	59.8±0.7	45.0	3.3±0.0	47.7	1.7±2.3	42.2
EGART Vote Sample ($\tau = 0.8$)	88.9±0.1	53.8	60.4±0.6	59.0	6.7±0.0	56.3	5.0±2.4	56.8
`Qwen2.5-7B-Instruct`								
CoT	64.7±0.4	1	57.2±0.3	1	8.4±2.3	1	5.0±2.4	1
CoT-SC	**89.4±0.0**	40	**74.4±0.3**	40	15.0±2.4	40	**18.3±2.4**	40
ToT Score TopK	80.5±0.3	45	60.0±0.1	45	**16.7±4.7**	45	8.4±2.3	45
EGART Score TopK ($\tau = 1.0$)	81.2±0.2	39.9	58.0±0.1	38.2	13.3±0.0	25.0	5.0±2.4	24.0
EGART Score TopK ($\tau = 0.9$)	80.6±1.1	41.8	59.6±0.9	40.1	10.0±4.7	28.3	6.7±4.7	27.3
EGART Score TopK ($\tau = 0.8$)	80.4±0.4	42.9	58.4±0.6	41.5	**16.7±4.7**	32.4	8.4±2.3	29.8
EGART Score TopK ($\tau = 0.7$)	81.4±0.4	43.7	59.5±1.1	42.3	13.3±0.0	36.6	6.7±0.0	34.5
ToT Score Sample	80.8±0.6	45	59.9±0.2	45	10.0±0.0	45	6.7±4.7	45
EGART Score Sample ($\tau = 1.0$)	79.9±0.6	39.9	59.0±0.1	38.3	15.0±7.1	25.6	1.7±2.3	25.7
EGART Score Sample ($\tau = 0.9$)	79.9±0.3	41.8	59.6±0.5	40.0	11.7±7.1	27.5	1.7±2.3	26.8
EGART Score Sample ($\tau = 0.8$)	81.0±0.5	43.0	58.5±0.4	41.4	11.7±2.3	33.5	6.7±0.0	31.6
EGART Score Sample ($\tau = 0.7$)	82.0±0.5	43.5	59.3±0.4	42.3	13.4±9.4	36.7	3.4±4.7	35.4
ToT Vote TopK	82.9±0.1	45	58.8±0.6	45	11.7±2.3	45	3.4±4.7	45
EGART Vote TopK ($\tau = 1.0$)	80.9±1.7	25.1	58.2±0.5	25.5	11.7±2.3	27.2	6.7±0.0	26.2
EGART Vote TopK ($\tau = 0.8$)	82.7±0.8	39.4	59.5±0.1	40.2	10.0±4.7	38.9	3.4±4.7	39.3
EGART Vote TopK ($\tau = 0.7$)	82.6±0.7	54.8	59.0±0.4	56.1	10.0±0.0	61.4	3.3±0.0	63.3
ToT Vote Sample	80.2±0.1	45	59.3±0.5	45	11.7±2.3	45	5.0±2.4	45
EGART Vote Sample ($\tau = 1.0$)	80.2±0.0	24.9	57.5±0.4	25.5	11.7±2.3	27.1	5.0±7.1	25.8
EGART Vote Sample ($\tau = 0.8$)	80.6±0.6	39.6	58.1±0.1	40.2	10.0±0.0	39.4	5.0±2.4	41.7
EGART Vote Sample ($\tau = 0.7$)	81.0±0.4	54.0	59.7± 1.5	56.3	8.4±2.3	56.5	3.4±4.7	59.0

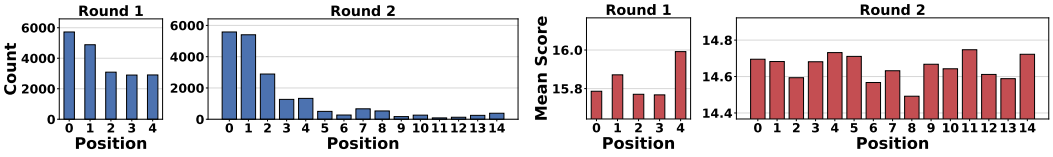


(a) Qwen2.5-3B GSM8K Vote Sample w/ Repl. (b) Qwen2.5-3B GSM8K Score Sample w/ Repl.

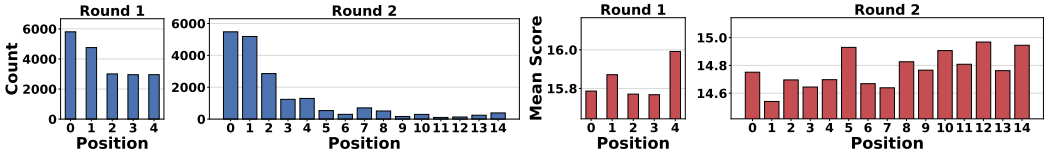


(c) Qwen2.5-3B GSM8K Vote Top-K (d) Qwen2.5-3B GSM8K Score TopK

Figure 12: Evaluation Behaviors of the Two Evaluation Strategies on GSM8K with Qwen2.5-3B.

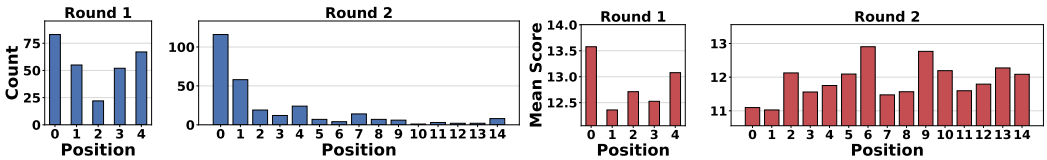


(a) Qwen2.5-3B MATH Vote Sample w/ Repl. (b) Qwen2.5-3B MATH Score Sample w/ Repl.

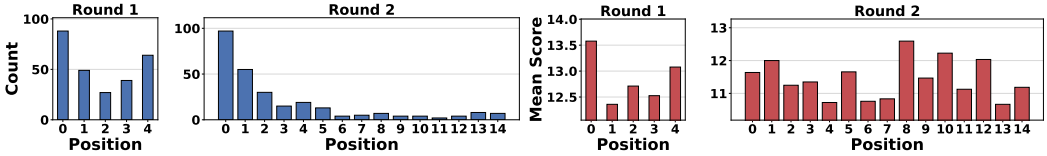


(c) Qwen2.5-3B MATH Vote Top-K (d) Qwen2.5-3B MATH Score TopK

Figure 13: Evaluation behaviors of the two evaluation strategies on MATH with Qwen2.5-3B.

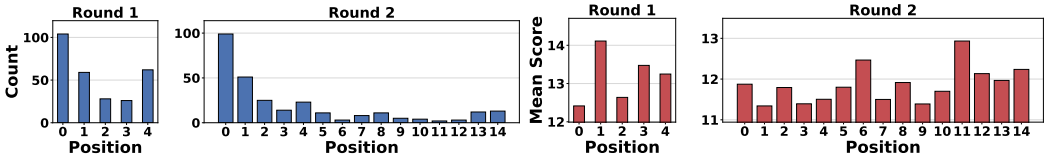


(a) Qwen2.5-3B AIME24 Vote Sample w/ Repl. (b) Qwen2.5-3B AIME24 Score Sample w/ Repl.

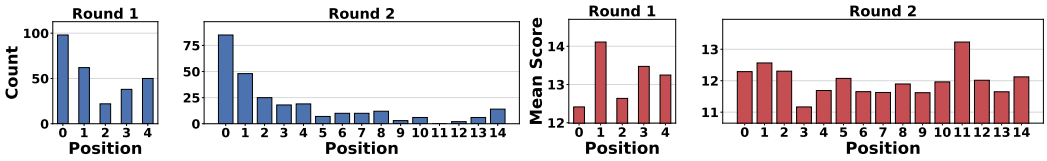


(c) Qwen2.5-3B AIME24 Vote Top-K (d) Qwen2.5-3B AIME24 Score TopK

Figure 14: Evaluation behaviors of the two evaluation strategies on AIME24 with Qwen2.5-3B.

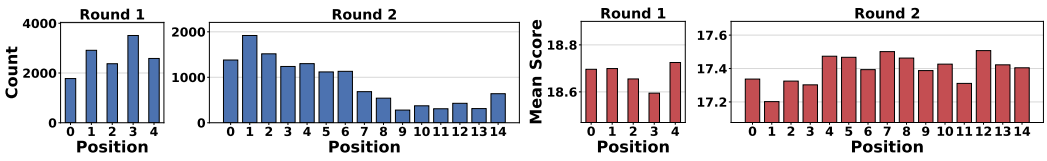


(a) Qwen2.5-3B AIME25 Vote Sample w/ Repl. (b) Qwen2.5-3B AIME25 Score Sample w/ Repl.

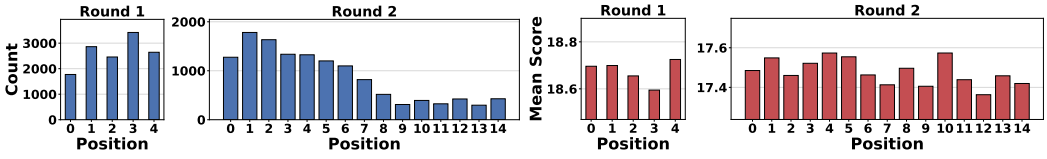


(c) Qwen2.5-3B AIME25 Vote Top-K (d) Qwen2.5-3B AIME25 Score TopK

Figure 15: Evaluation behaviors of the two evaluation strategies on AIME25 with Qwen2.5-3B.

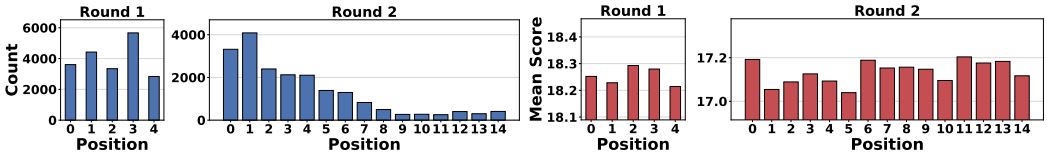


(a) Qwen2.5-7B GSM8K Vote Sample w/ Repl. (b) Qwen2.5-7B GSM8K Score Sample w/ Repl.

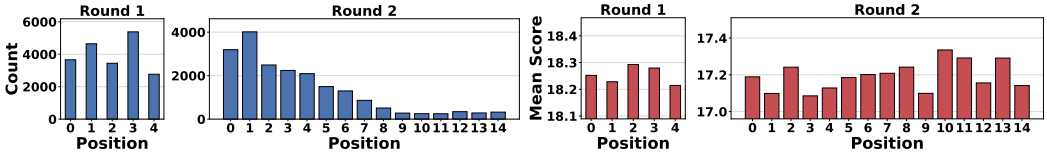


(c) Qwen2.5-7B GSM8K Vote Top-K (d) Qwen2.5-7B GSM8K Score TopK

Figure 16: Evaluation behaviors of the two evaluation strategies on GSM8K with Qwen2.5-7B.

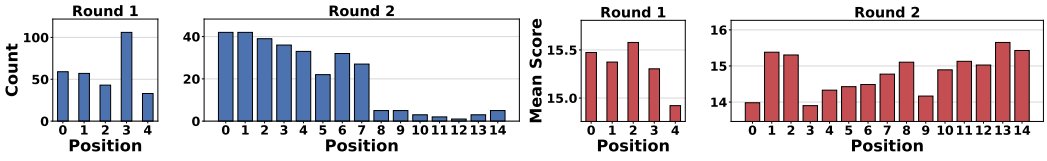


(a) Qwen2.5-7B MATH Vote Sample w/ Repl. (b) Qwen2.5-7B MATH Score Sample w/ Repl.

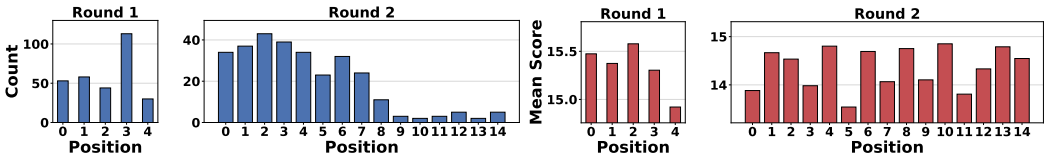


(c) Qwen2.5-7B MATH Vote Top-K (d) Qwen2.5-7B MATH Score TopK

Figure 17: Evaluation behaviors of the two evaluation strategies on MATH with Qwen2.5-7B.

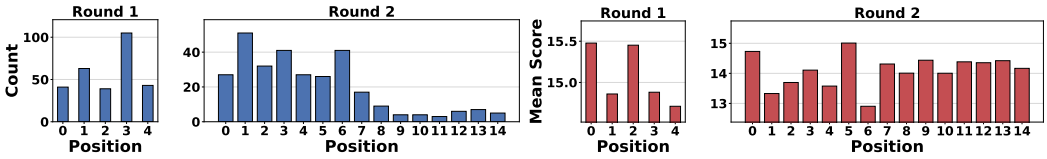


(a) Qwen2.5-7B AIME24 Vote Sample w/ Repl. (b) Qwen2.5-7B AIME24 Score Sample w/ Repl.

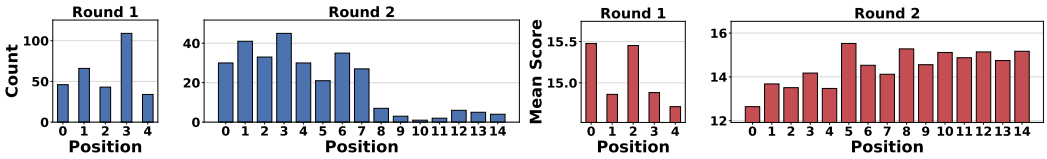


(c) Qwen2.5-7B AIME24 Vote Top-K (d) Qwen2.5-7B AIME24 Score TopK

Figure 18: Evaluation behaviors of the two evaluation strategies on AIME24 with Qwen2.5-7B.



(a) Qwen2.5-7B AIME25 Vote Sample w/ Repl. (b) Qwen2.5-7B AIME25 Score Sample w/ Repl.



(c) Qwen2.5-7B AIME25 Vote Top-K (d) Qwen2.5-7B AIME25 Score TopK

Figure 19: Evaluation behaviors of the two evaluation strategies on AIME25 with Qwen2.5-7B.