# Not All Queries Need Deep Thought: CoFiCot for Adaptive Coarse-to-fine Stateful Refinement

**Dongxu Zhang**[1,2,*] **Hongqiang Lin**[3,*] **Yiding Sun**[1,2,*] **Pengyu Wang**[4]

**Qirui Wang**[1] **Ning Yang**[5,†] **Jihua Zhu**[1,2,†]

[1]School of Software Engineering, Xi'an Jiaotong University

[2]State Key Laboratory of Integrated Services Networks, Xidian University [3]Zhejiang University

[4]The Chinese University of Hong Kong (Shenzhen) [5]Institute of Automation, Chinese Academy of Sciences

## Abstract

Scaling test-time computation enhances LLM reasoning ability but faces a uniform computation paradox. Allocating identical resources leads to over-correction on simple tasks and insufficient refinement on complex ones. To address this, we propose CoFiCot, a coarse-to-fine adaptive framework that dynamically tailors inference strategies to problem difficulty. Specifically, we implement a multi-metric classifier that triages queries by synthesizing semantic entropy, consensus reliability, and predicted reasoning depth . This enables a differentiated refinement stage that applies efficient aggregation for simple queries while routing complex ones to a context-aware correction loop . We formalize correction as a stateful sequential propagation process , where each repair is strictly conditioned on the verified history of prior rectifications. By integrating Process Reward Models (PRMs) within this state-dependent trajectory, CoFiCot effectively bridges the gap between granular error localization and global logical coherence, preventing the context fragmentation typical of stateless refinement methods.

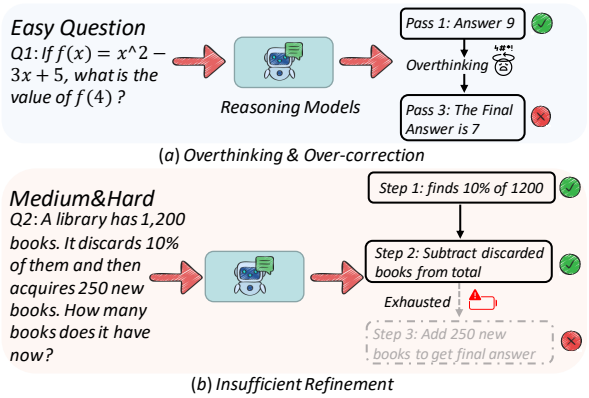


Figure 1: Conceptual illustration of the fundamental paradox of uniform computation in LLM reasoning. (Top) For an *Easy* Question, the model correctly computes the answer in its initial pass but is forced into unnecessary iterations. This overthinking leads to over correction, where the correct answer is corrupted into a final incorrect one. (Bottom) For a *Medium&Hard* Question, the same fixed computational budget is insufficient. The reasoning process is prematurely terminated before all logical steps are completed, resulting in an insufficient refinement failure.

## 1 INTRODUCTION

The advent of Chain of Thought (CoT) prompting has endowed Large Language Models (LLMs) with the ability to deconstruct complex problems into intermediate steps [Kojima et al., 2022, Zhang et al., 2026b]. Following this trajectory, recent advances like OpenAI's o1 [Jaech et al., 2024] and DeepSeekR1 [Guo et al., 2025] have empirically validated the test-time scaling law, demonstrating that allocating additional compute to extended reasoning trajectories yields performance gains comparable to parameter scaling. As a result, optimizing test-time computation has emerged as the pivotal frontier for next generation reasoning [Snell et al., 2024]. Methods such as Self-Consistency [Wang et al., 2022] and Best-of-$k$ [Sun et al., 2024] leverage this by generating and aggregating diverse reasoning paths.

However, allocating identical computational resources to every query is fundamentally misaligned with the variable complexity of reasoning tasks, as real world problems demand diverse levels of cognitive effort [Wang et al., 2023a, Yang et al., 2025a]. This misalignment manifests differently across difficulty levels. For simple tasks, forcing extended reasoning not only causes rapid performance saturation [Wadhwa et al., 2024], but also leads to hallucination.

[*]Equal contribution.

[†]Corresponding authors.

*Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence* (UAI 2026), PMLR 337:8115–8129.

For complex tasks, this uniform strategy fails to address the inherent brittleness of reasoning chains, where a single error can invalidate the entire process [Lightman et al., 2023]. In light of this, current LLMs' inability to emulate this adaptive allocation results in the critical uniform computation paradox, characterized by the dual failure modes of over thinking simple queries and under refining complex ones, as illustrated in Figure 1.

To resolve the paradox of uniform allocation, the field is shifting towards adaptive computation [Zhang et al., 2025, Huang et al., 2023], aiming to tailor inference workflows dynamically. However, current approaches are often specialized, failing to bridge the gap between efficiency and robust error recovery. Although routing strategies [Xia et al., 2025] improve efficiency, they lack the mechanisms to actively repair reasoning errors. Furthermore, iterative refinement methods [Madaan et al., 2023] often function in a stateless manner, leading to the phenomenon where correcting an intermediate step invalidates the subsequent logical flow.

Drawing inspiration from effective human problem solving, which relies on metacognitive triage to allocate minimal resources to straightforward tasks while reserving iterative effort for complex ones [Zhang et al., 2026c, Sun et al., 2026], we introduce CoFiCot, a Coarse-to-fine framework designed to resolve the paradox of uniform computation. Our method operates via a two stage pipeline. First, a coarse-grained classification stage acts as a lightweight semantic router which triages problems into *Easy*, *Medium*, or *Hard*. Second, a fine-grained refinement stage executes a differentiated strategy. *Easy* problems are resolved via efficient aggregation, while *Medium* and *Hard* problems enter an iterative correction loop for further refinement.

Crucially, we introduce a Stateful Sequential Correction mechanism designed to enforce logical consistency during the repair process. Unlike prevailing methods that regenerate entire chains [Madaan et al., 2023], our mechanism formalizes correction as a history generative process. By anchoring the validated reasoning path and initiating a new decoding branch from the point of error, we ensure that corrections trigger a cascading update of all dependent reasoning steps, reconciling the precision of step-level error localization with the coherence required for complex reasoning.

Our contributions are summarized as follows:

- We propose CoFiCot, an adaptive framework that dynamically matches reasoning strategies according to the problem difficulty.
- We devise a sequential correction mechanism for complex problems. Our approach treats reasoning as a state-dependent trajectory, ensuring corrections propagate downstream.
- Extensive experiments across seven benchmarks demonstrate that CoFiCot outperforms strong baselines and achieves a better trade-off between accuracy and efficiency.

## 2 RELATED WORK

### 2.1 ENHANCING REASONING ABILITY IN LLMS

Recent research in LLM reasoning has shifted focus from optimizing CoT prompting [Kojima et al., 2022] to ensemble based inference strategies [Zhang et al., 2026a]. To mitigate the fragility of single reasoning paths, aggregation strategies like Self-Consistency [Wang et al., 2022] and Best-of-$k$ [Lightman et al., 2023] generate multiple candidates to select a consensus or high reward answer. However, these methods suffer from performance saturation, where computational costs rise linearly while accuracy gains plateau [Li et al., 2024, Yin et al., 2024].

Parallel to these efforts, Iterative Refinement has emerged as a promising paradigm, allowing models to actively improve outputs [Madaan et al., 2023]. Yet, indiscriminate refinement causes over correction on simple problems, corrupting correct answers [Chen et al., 2024b, Liu et al., 2024] while providing insufficient rectification for complex reasoning failures [Li et al., 2024]. These challenges underscore that simply iterating is insufficient.

### 2.2 COMPUTATIONAL EFFICIENCY IN LLMS

The high computational cost of CoT has driven research into efficiency via compression techniques like token pruning [Xia et al., 2025] or step reduction [Kumar et al., 2025]. While these methods mitigate per-token costs, they often remain limited by static inference paths that fail to account for varying query complexities [Schwartz et al., 2020]. A more promising paradigm is Adaptive Computation, where Coarse-to-fine (C2F) frameworks attempt to tailor resource allocation dynamically [Wang et al., 2023b]. However, current C2F solutions often face a trade-off between latency and corrective capability. Methods like MAgICoRe [Chen et al., 2024a] incur prohibitive latency, whereas early exit strategies [Zhang et al., 2025] lack mechanisms to actively repair flawed reasoning. CoFiCot bridges this gap by unifying granular difficulty assessment with stateful refinement to achieve an optimal balance between accuracy and efficiency.

### 2.3 FEEDBACK-DRIVEN ERROR RECTIFICATION

The efficacy of iterative refinement hinges on feedback quality [Chen et al., 2024b, Wang et al., 2022]. Intrinsic self-correction often fails, as models struggle to self-diagnose errors and are prone to generating invalid solutions [Cao et al., 2025, Yang et al., 2025b]. Accordingly, research has

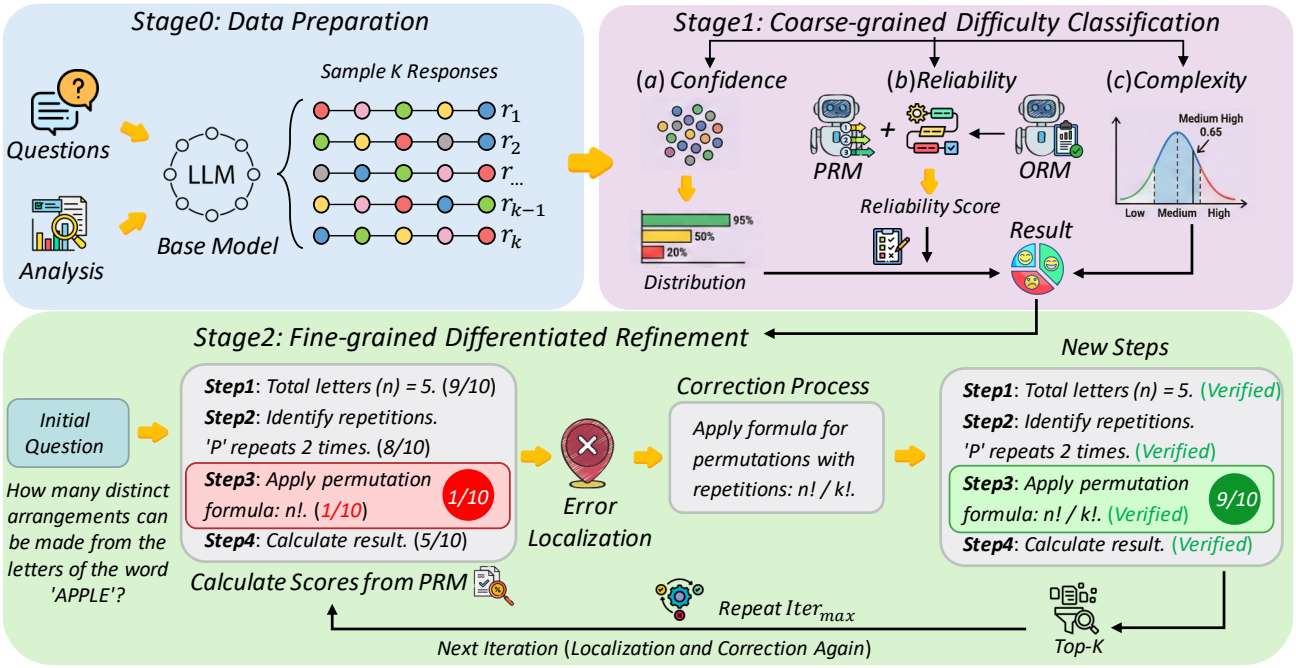


Figure 2: The complete workflow of the CoFiCot framework. The process begins in Stage 0, where the base LLM generates an initial ensemble of $k$ reasoning traces. This set is passed to Stage 1, which performs a parallel, multi metric analysis to assess difficulty. *Easy* problems are resolved with simple aggregation. *Medium* and *Hard* problems are channeled into Stage 2. This stage initiates an iterative loop. A PRM scores each step, identifying a flawed step, which is then fed into the Correction Process. Critically, this correction is context aware, conditioning on the history of previous, correct steps to generate a new solution. The refined solution is then selected via an ORM, and the process repeats until termination criteria are met.

pivoted toward external feedback via specialized Reward Models (RMs) [Li et al., 2024]. Although Outcome RMs (ORMs) evaluate entire reasoning traces [Ma et al., 2025, Lu et al., 2024], Process RMs (PRMs) provide dense, step-level signals [Zhu et al., 2025], current integration methods often results in a lack of global coherence. CoFiCot addresses this limitation by formalizing correction as a propagation process, ensuring that RM-guided repairs maintain the integrity of the entire reasoning.

# 3 METHODOLOGY

## 3.1 OVERVIEW AND DATA PREPARATION

CoFiCot operates through a tiered inference pipeline that begins by constructing a diverse solution space to facilitate subsequent complexity analysis. During the data preparation stage, we initialize the workflow by generating an ensemble of $k$ candidate solutions sampling from a base LLM. The resulting set denoted as $\mathcal{R}_0 = \{r_1, \ldots, r_k\}$, consists of independent CoT reasoning traces. Each solution $r_i \in \mathcal{R}_0$ represents a self contained attempt at solving the problem. This solution set serves as the foundation for the Coarse-to-fine procedure, where the coarse-grained classification stage acts as an analytical routing mechanism to evaluate problem difficulty. Based on this assessment, queries are either resolved via immediate aggregation or channeled into a fine-grained refinement stage featuring an iterative correction loop for more challenging cases. The complete workflow is illustrated in Figure 2, with formal execution details provided in Algorithm 1 in the Appendix A.

## 3.2 STAGE 1: COARSE-GRAINED CLASSIFICATION

This stage performs a systematic analysis of the initial solution set $\mathcal{R}_0$ to classify each problem into one of three categories: *Easy*, *Medium* or *Hard*. The classification is derived from a synthesis of three complementary metrics.

**Confidence Assessment** This metric gauges predictive uncertainty via the semantic consensus of $\mathcal{R}_0$. We partition the generated solutions into $n$ semantic clusters $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ and define a reward probability for the $i$-th cluster:

$$p(\mathcal{A}_i) = \frac{\sum_{r_j \in \mathcal{A}_i} \mathcal{S}_j}{\sum_{m=1}^{n} \sum_{r_l \in \mathcal{A}_m} \mathcal{S}_l}, \tag{1}$$

where $\mathcal{S}_j$ denotes the RM score of solution $r_j$. We quantify

the predictive uncertainty via the Shannon entropy $H(\mathcal{A})$ of the answer cluster distribution:

$$H(\mathcal{A}) = -\sum_{i=1}^{n} p(\mathcal{A}_i) \log p(\mathcal{A}_i). \tag{2}$$

To facilitate thresholding, we invert and normalize $H(\mathcal{A})$ into a confidence score $C \in [0, 1]$ via a scaled sigmoid function:

$$C = \sigma(\alpha \cdot (1 - H(\mathcal{A}))), \tag{3}$$

where $\alpha$ is a scaling hyperparameter. The preliminary difficulty label $d_1$ is then assigned based on empirical thresholds:

$$d_1 = \begin{cases} \textit{Easy} & \text{if } C > 0.6 \\ \textit{Medium} & \text{if } 0.3 < C \leq 0.6 \\ \textit{Hard} & \text{if } C \leq 0.3 \end{cases}. \tag{4}$$

The low entropy typically signifies a distribution sharply peaked around a consensus, we acknowledge that high confidence does not guarantee correctness due to potential confident hallucination. Therefore, this metric serves as a preliminary filter and is coupled with the Reliability to validate the consensus quality (detailed in Appendix B).

**Reliability Assessment** This metric contextualizes the consensus quality to filter out false consensus scenarios. We identify the majority answer cluster $\mathcal{A}_g$, compute its average RM score $\bar{\mathcal{S}}_g$, and normalize it against the global score distribution of $\mathcal{R}_0$ via a Z-score transformation:

$$\mathcal{S}_{\text{norm}} = \frac{\bar{\mathcal{S}}_g - \mu_{\text{all}}}{\sigma_{\text{all}}}, \tag{5}$$

where $\mu_{\text{all}}$ and $\sigma_{\text{all}}$ represent the global mean and standard deviation. A negative $\mathcal{S}_{\text{norm}}$ indicates that the popular consensus underperforms the candidate pool's average quality. Accordingly, we map this score to the second difficulty label $d_2$:

$$d_2 = \begin{cases} \textit{Easy} & \text{if } \mathcal{S}_{\text{norm}} \geq \delta \\ \textit{Medium} & \text{if } 0 \leq \mathcal{S}_{\text{norm}} < \delta \\ \textit{Hard} & \text{if } \mathcal{S}_{\text{norm}} < 0 \end{cases}, \tag{6}$$

where $\delta$ is a positive threshold that delimits high quality consensus. This ensures that answers with significant quality superiority are trusted as *Easy*.

**Complexity Assessment** Finally, we estimate the intrinsic problem complexity by prompting the base LLM to predict the necessary reasoning steps:

$$N_{\text{steps}} = \mathcal{G}_{\text{llm}}(Q \oplus P_{\text{predict}}). \tag{7}$$

This step incurs negligible overhead as it requires generating only a single integer token. To determine the difficulty label $d_3$, we map $N_{\text{steps}}$ against the inverse CDF $F_L^{-1}$ of solution lengths from a representative corpus, utilizing tertile based thresholds (details provided in Appendix B):

$$d_3 = \begin{cases} \textit{Easy} & \text{if } N_{\text{steps}} \leq F_L^{-1}(0.33) \\ \textit{Medium} & \text{else} \\ \textit{Hard} & \text{if } N_{\text{steps}} > F_L^{-1}(0.67) \end{cases}. \tag{8}$$

**Final Difficulty Synthesis** To obtain the final classification, we employ a Balanced Strategy. The final difficulty score is a weighted sum of the mapped metric values:

$$D_{\text{score}} = \sum_{i=1}^{3} w_i \cdot \text{val}(d_i), \tag{9}$$

where $d_i$ denotes the difficulty label assigned by the $i$-th metric, $\text{val}(\cdot)$ represents the mapping function that transforms each qualitative difficulty label into a discrete numerical rank and $w_i$ is the scalar weight reflecting the relative importance of each metric. We provide a detailed justification for these hyperparameters and an analysis of their sensitivity in Appendix B.

### 3.3 STAGE 2: FINE-GRAINED REFINEMENT

Upon determining the difficulty of the problem $D_{\text{final}}$, the framework executes a differentiated refinement strategy. *Easy* problems bypass the expensive refinement loop and are resolved on the initial ensemble $\mathcal{R}_0$ (details in Appendix E).

Problems classified as *Medium* or *Hard* are subjected to an iterative refinement loop. Let $\mathcal{R}_{t-1}$ be the set of $k$ candidate solutions at the beginning of iteration $t$. The refinement process employs a sequential correction mechanism, modeled as a state-dependent transformation function $\Phi$. For each solution $r_i \in \mathcal{R}_{t-1}$ composed of steps $\{s_{i,1}, s_{i,2}, \dots\}$, a PRM provides step-level scores. Steps falling below a threshold $\tau_{\text{step}}$ are flagged as errors. Upon detecting the first error step $s_{i,j}$, we freeze the verified history $H_{i,j-1}^{(t)}$ and invoke $\Phi$ (details in Appendix B) to generate a corrected step. To ensure the correction logically propagates, the generation is conditioned on the original question $Q$ and the history:

$$s_{i,j}^{(t)} = \Phi\left(Q, s_{i,j}^{(t-1)}, \mathcal{F}_{i,j}, H_{i,j-1}^{(t)}\right), \tag{10}$$

where $Q$ is the problem statement, $s_{i,j}^{(t-1)}$ is the original erroneous step, $\mathcal{F}_{i,j}$ is the PRM derived feedback, and $H_{i,j-1}^{(t)} = \{s_{i,1}^{(t)}, \dots, s_{i,j-1}^{(t)}\}$ is the sequence of verified steps preceding the error.

Crucially, unlike stateless methods that simply replace the error step, our mechanism adheres to causal consistency. Verified steps before the error are carried over unchanged ($s_{i,k}^{(t)} = s_{i,k}^{(t-1)}$ for $k < j$), while all steps following the correction are regenerated based on the updated state. The

refined solution $r_i^{(t)}$ is thus a coherent sequence assembled from the preserved history and the corrected step.

Following the correction of all $k$ solutions, an iterative selection and evaluation process is employed to curate the most promising candidates for the subsequent iteration. Let $\mathcal{R}^{(t)} = \{r_1^{(t)}, \ldots, r_k^{(t)}\}$ be the set of $k$ newly refined solutions. This set is merged with the original set $\mathcal{R}_{t-1}$ to form an expanded candidate pool:

$$\mathcal{P}_t = \mathcal{R}_{t-1} \cup \mathcal{R}^{(t)}. \tag{11}$$

An ORM then provide a quality score for each solution in this pool. The top-$k$ solutions that maximize the aggregate quality are selected to form the input set for the next iteration $\mathcal{R}_t$, by solving the following optimization problem:

$$\mathcal{R}_t = \underset{\mathcal{S} \subseteq \mathcal{P}_t, |\mathcal{S}|=k}{\arg\max} \sum_{r \in \mathcal{S}} \text{ORM}(r). \tag{12}$$

This iterative cycle of correction and selection is governed by a dynamic termination protocol executed at the end of each iteration. The expanded solution set $\mathcal{R}_t$ is reevaluated by the Stage 1 classifier. We also employ an Early Exit strategy. If the difficulty downgrades to *Easy*, the loop terminates immediately to save compute. Otherwise, the refinement continues until the predefined budget ($Iter_{max}$) is exhausted (details in Appendix E).

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

We outline the experimental setup spanning seven diverse benchmarks and two backbone models. Furthermore, we compare against strong baselines to rigorously evaluate the trade-off between accuracy and efficiency of our CoFiCot.

**Datasets and Evaluation Metrics** We conduct our evaluation across seven challenging datasets, which can be grouped into mathematical reasoning and general reasoning [Wang et al., 2025]. We use a diverse suite of five reasoning benchmarks. This includes GSM8K [Cobbe et al., 2021] and SVAMP [Patel et al., 2021], which are math word problems; MATH [Hendrycks et al., 2021], a benchmark of high school competition level mathematics problems; and MMLU [Hendrycks et al., 2020] and SAT [Zhong et al., 2023], which are standard mathematical reasoning subsets. To test the generalization of our framework beyond mathematical domains, we also evaluate on two additional tasks: ARC [Clark et al., 2018], a complex commonsense reasoning dataset and Date [Srivastava et al., 2023], which requires logical reasoning about dates. Our evaluation is based on two primary categories of metrics. The primary metric is the final accuracy of the model. And we also report the average number of samples generated per question denoted as $k$, which accounts for the adaptive nature of our refinement loop and the total number of decoded tokens.

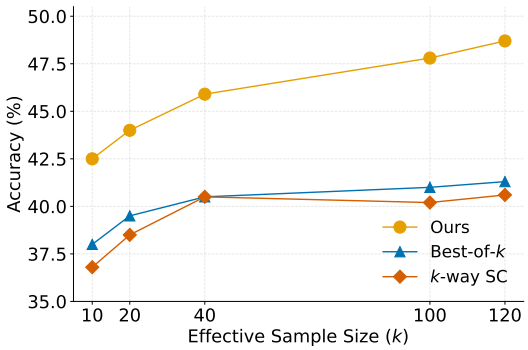


Figure 3: Accuracy vs. Effective Sample Size ($k$) on the MATH dataset.

**Models and Implementation Details** All experiments are conducted using Llama-3-8B-Instruct [Dubey et al., 2024] and GPT-3.5-Turbo [Schulman et al., 2022] to ensure our findings are generalizable. Our framework leverages two pretrained RMs. We employ InternLM-7B as ORM and Math-Shepherd-7B as PRM. For all experiments, we generate 40 initial candidate solutions using a decoding temperature of 0.8 to ensure diversity. For our coarse stage, we apply uniform weights ($w_1 = w_2 = w_3 = 1/3$) to the scores from our three classification metrics. For the fine stage, we set the step-level verification threshold $\tau_{step} = 0.5$, and the maximum number of iterations ($Iter_{max}$) is set to 2 for *Medium* problems and 3 for *Hard* problems. For hyperparameters, we set the entropy scaling factor $\alpha = 2$ and the Z-score threshold $\delta = 0.5$.

**Baselines for Comparison** We compare CoFiCot against with other strong baselines. Vanilla Prompting [Kojima et al., 2022] is a standard zero-shot CoT prompt, which generates a single reasoning chain. Aggregation based Methods [Yin et al., 2024] generate a large number of samples and aggregate them. We compare against $k$-way Self-Consistency [Li et al., 2023], which uses a majority vote and Best-of-$k$ [Lightman et al., 2023], which uses the ORM to select the best solution. To establish a performance ceiling, these baselines are given a high budget of $k = 120$ samples.

## 4.2 MAIN RESULTS

We present a comprehensive empirical evaluation of CoFiCot against strong baselines across two primary domains: mathematical reasoning and general commonsense reasoning.

**Performance on Mathematical Reasoning** The primary results of our evaluation are presented in Table 1. The findings clearly demonstrate that CoFiCot consistently and significantly outperforms all baseline methods. On the Llama-

| Method | MMLU | MATH | SVAMP | GSM8K | SAT | Avg. |
|---|---|---|---|---|---|---|
| *Llama3-8B-Instruct* | | | | | | |
| Zero-shot CoT [Wei et al., 2022] | 50.4 | 24.2 | 72.4 | 80.1 | 58.2 | 57.1 |
| Self-Refine [Madaan et al., 2023] | 49.6 | 24.6 | 72.0 | 79.0 | 57.7 | 56.6 |
| Best-of-$k$ ($k$ = 120) [Lightman et al., 2023] | 62.6 | 41.4 | 88.7 | 90.1 | 72.4 | 71.0 |
| $k$-way SC ($k$ = 120) [Wang et al., 2022] | 63.0 | 40.6 | 89.8 | 90.3 | 70.5 | 70.8 |
| Self-Refine + $k$-way SC [Chen et al., 2024a] | 62.1 | 40.4 | 88.6 | 90.1 | 68.2 | 69.9 |
| CoFiCot(ours) | **68.8** | **47.9** | **91.4** | **91.8** | **75.0** | **75.0** |
| *GPT-3.5-Turbo* | | | | | | |
| Zero-shot CoT [Wei et al., 2022] | 62.5 | 37.2 | 78.1 | 78.5 | 76.8 | 66.6 |
| Self-Refine [Madaan et al., 2023] | 62.4 | 37.4 | 77.7 | 77.4 | 77.3 | 66.4 |
| Best-of-$k$ ($k$ = 120) [Lightman et al., 2023] | 70.1 | 50.6 | 87.7 | 90.5 | 87.8 | 77.3 |
| $k$-way SC ($k$ = 120) [Wang et al., 2022] | 70.4 | 51.2 | 86.9 | 89.8 | 87.6 | 77.2 |
| Self-Refine + $k$-way SC [Chen et al., 2024a] | 69.4 | 49.8 | 86.9 | 88.1 | 85.6 | 76.0 |
| CoFiCot(ours) | **73.5** | **57.7** | **89.8** | **91.2** | **90.1** | **80.5** |

Table 1: Performance comparison of our proposed CoFiCot method against baseline methods on the Llama3-8B-Instruct and GPT-3.5-Turbo models. Results are reported across five reasoning benchmarks. Best performance is highlighted in bold. We report accuracy(%).

| Method | ARC | Date |
|---|---|---|
| Zero-shot [Wei et al., 2022] | 66.5 | 52.5 |
| 40-way SC [Wang et al., 2022] | 85.5 | 72.5 |
| 120-way SC [Chen et al., 2024a] | 86.0 | 72.5 |
| CoFiCot(ours) | **88.2** | **80.8** |

Table 2: Performance comparison on commonsense reasoning and logical reasoning benchmarks. We also report accuracy(%).

3-8B-Instruct model, CoFiCot achieves an average accuracy of 75.0%, representing a substantial 4.0% absolute improvement over the strongest baseline Best-of-$k$. Notably, CoFiCot achieves a 6.5% absolute gain over the baseline's 41.4% on MATH. This trend is consistent on the GPT-3.5-Turbo model, where CoFiCot achieves an average accuracy of 80.5%, surpassing the strongest baseline by 3.2%.

A hypothesis of our work is that brute force aggregation methods suffer from performance saturation. Figure 3 provides a clear visualization of this phenomenon on the MATH dataset. The accuracy of both $k$-way SC and Best-of-$k$ grows minimally as the sample size increases from $k = 40$ to $k = 120$, quickly hitting a performance plateau. In contrast, CoFiCot not only starts from a significantly higher baseline accuracy but also scales more effectively with an increased computational budget. Most notably, CoFiCot initiated with an effective sample size of $k = 40$ achieves 45.9% accuracy that is already superior to both $k$-way SC (40.6%) and Best-of-$k$ (41.4%) using a massive $k = 120$ budget.

**Generalization to Commonsense Reasoning** To demonstrate the general domain applicability of our framework, we evaluated CoFiCot on the ARC [Clark et al., 2018] and Date [Srivastava et al., 2023] datasets. As shown in Table 2, CoFiCot's superiority is not limited to mathematical tasks. Our method achieves an accuracy of 88.2% in ARC, surpassing the baseline of SC 120-ways by 2.2%. The improvement is more pronounced on the Date Understanding task, where CoFiCot (80.8%) outperforms 120-way SC (72.5%) by a significant 8.3%.

### 4.3 ADDITIONAL ANALYSIS

To further probe the robustness and characteristics of our CoFiCot framework, we conduct a series of additional analyses covering scalability, modularity, and other critical aspects (detailed in Appendix C).

**Performance Analysis with Different Models** A key question is whether CoFiCot's benefits are limited to improving weaker base models. To test this, we apply our framework to Qwen2.5-Math-7B, a model for mathematical reasoning. The results presented in Table 3 demonstrate that CoFiCot continues to provide performance gains.

CoFiCot achieves an average accuracy of 93.4%. The performance gains are particularly pronounced on the dataset MATH, where CoFiCot (91.7%) substantially outperforms the baselines. This result confirms that CoFiCot is not merely a compensatory mechanism but an effective framework for scaling and enhancing the reasoning capabilities.

The performance of CoFiCot is guided by its external ORM and PRM. In Table 4, we analyze the modularity of our framework by swapping these components on the MATH dataset. The results clearly indicate that CoFiCot's performance scales directly with the quality of its components.

| Method | MMLU | MATH | SVAMP | GSM8K | SAT | Avg. |
|---|---|---|---|---|---|---|
| Qwen2.5-Math-7B [Yang et al., 2024] | 73.9 | 78.8 | 91.8 | 94.9 | 92.3 | 86.3 |
| $k$-way SC ($k$ = 40) [Wang et al., 2022] | 81.3 | 87.0 | 95.5 | 97.2 | 97.3 | 91.7 |
| $k$-way SC ($k$ = 120) [Chen et al., 2024a] | 82.0 | 86.8 | 95.4 | **97.3** | 97.3 | 91.8 |
| Best-of-$k$ ($k$ = 120) [Lightman et al., 2023] | 82.6 | 86.0 | 93.4 | 96.9 | 95.2 | 90.8 |
| CoFiCot(ours) | **84.9** | **91.7** | **95.8** | 97.2 | **97.6** | **93.4** |

Table 3: Evaluating CoFiCot's scalability on the powerful Qwen2.5-Math-7B model. We report accuracy(%).

| PRM | ORM | Acc. |
|---|---|---|
| Math-Shepherd-7B | InternLM-7B | 48.3 |
| Qwen-Math 7B | InternLM-7B | 54.9 |
| Math-Shepherd-7B | Llama-3.1-8B | 51.2 |

Table 4: Analysis of CoFiCot's sensitivity to the quality of its component reward models. Accuracy (Acc. %) is shown for different PRM and ORM combinations.

| Methods | GSM8K | MATH |
|---|---|---|
| | Accuracy | Accuracy |
| CoFiCot | **91.8** | **47.9** |
| w/o Coarse Stage | $88.5_{(3.3\downarrow)}$ | $45.1_{(2.8\downarrow)}$ |
| w/o Fine Stage | $89.1_{(2.7\downarrow)}$ | $41.2_{(6.7\downarrow)}$ |

Table 5: Ablation study of the core *Coarse* and *Fine* stages of our CoFiCot framework on MATH and GSM8K. We report accuracy(%).

Upgrading the PRM from Math-Shepherd-7B to the stronger Qwen-Math 7B yields the most substantial accuracy boost, from 48.3% to 54.9% (+6.6%). Similarly, upgrading the ORM from InternLM-7B to Llama-3.1-8B also provides a significant gain (48.3% to 51.2%). This experiment validates the modular design of CoFiCot, demonstrating its ability to readily incorporate and benefit from advancements in reward models.

**Token Budget Analysis** While our main results demonstrated CoFiCot's superior sample efficiency, we provide a more granular analysis of the performance to cost trade-off in Figure 4. This figure plots accuracy against the total normalized token consumption for CoFiCot and the SC baselines across all five reasoning datasets.

CoFiCot achieves higher accuracy while consuming fewer or comparable tokens than the computationally expensive 120-way SC baseline. On MATH, CoFiCot achieves 47.9% accuracy using fewer tokens than 120-way SC. On MMLU and SAT, CoFiCot uses a fraction of the tokens consumed by 120-way SC to achieve a significantly higher accuracy. This analysis confirms that our adaptive method avoids the brute force cost of uniform sampling and provides a superior efficiency to performance ratio across the board.

**Latency vs. Trade-off** It is worth noting that while CoFiCot significantly reduces total token volume, the sequential nature of the Stage 2 refinement loop may incur higher latency per token compared to parallel sampling methods. Since CoFiCot bypasses the refinement loop for a large portion of *Easy* queries (approx. 40-60%), the average latency remains competitive. Furthermore, the reduction in computational load results in lower deployment costs and consumption (detailed in Appendix C).

### 4.4 CASE STUDY

To provide qualitative insight into the internal mechanics of our framework, we trace the full pipeline of CoFiCot using the "APPLE" permutation problem, as illustrated in Figure 2. This case study begins in Stage 0, where the base model generates an initial set of $k$ solutions. For this problem, many initial solutions are flawed, typically coalescing around the incorrect answer "120" by naively applying a simple $n!$ formula. This high prevalence of flawed solutions results in low answer quality and high uncertainty, causing our Stage 1 classifier to correctly categorize the problem as *Hard* and subsequently engage the fine-grained refinement loop. Once in stage 2, a flawed reasoning chain is submitted for correction and the synergy of our components becomes evident. The PRM first evaluates the chain step by step. It correctly identifies the source of the error by assigning high scores to Step 1 and Step 2 but a critically low score to the erroneous Step 3. This precise error localization triggers the generation of targeted feedback. The most critical stage is the subsequent context aware correction. The model is prompted to generate a new step, but is provided not only with the targeted feedback but also with the history of previously validated steps (Step 1 and Step 2). This contextual information is vital, as it allows the model to generate a Newstep3. Apply formula with repetitions: n! / k!. Furthermore, this correction propagates, enabling the model to generate a correct Newstep4: 120 / 2 = 60. Finally, this newly generated fully correct solution $r_i^{(t)}$ is returned to the candidate pool. The ORM evaluates this complete solution, assigns it

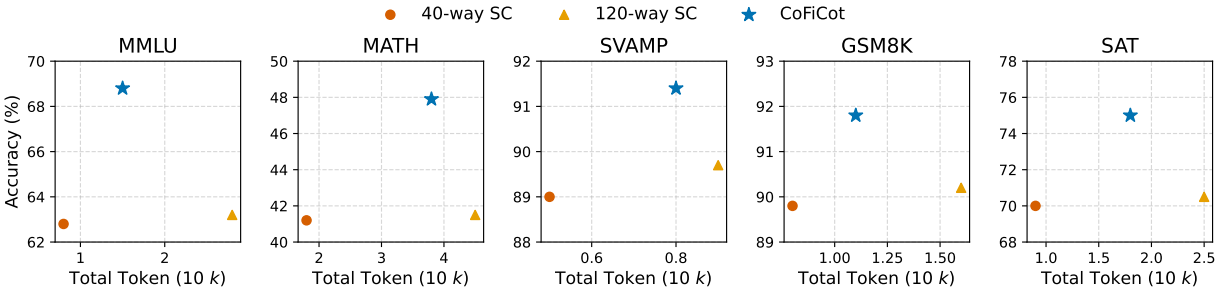


Figure 4: Token count and accuracy comparison across different datasets. The reported token count for CoFiCot includes the overhead from the initial sampling, Stage 1 classification tokens, and all PRM/ORM verification steps in Stage 2. While scaling Self-Consistency from $k = 40$ to $k = 120$ introduces substantial token overhead, our CoFiCot achieves significantly higher accuracy while being more token efficient than the 120-way SC baseline.

| Module | | | Datasets | | | |
|---|---|---|---|---|---|---|
| Cofi. | Relia. | Comp. | GSM8K | MATH | MMLU | SAT |
| ✓ | - | - | 86.8 | 43.4 | 63.9 | 71.2 |
| - | - | ✓ | 87.2 | 44.6 | 64.1 | 70.9 |
| - | ✓ | ✓ | 89.7 | 45.9 | 67.0 | 73.4 |
| ✓ | - | ✓ | 89.3 | 46.2 | 67.4 | 74.1 |
| ✓ | ✓ | ✓ | **91.8** | **47.9** | **68.8** | **75.0** |

Table 6: Ablation experiments of the CoFiCot. The results are measured in %. We also assess the impact of including (✓) or excluding (-) our Coarse stage evaluation metrics, Confidence Assessment(Cofi.) module, Reliability Assessment(Relia.) module and Complexity Assessment(Comp.) module.

a high holistic score, and it is thus selected as part of the new top-$k$ set for the next iteration.

### 4.5 ABLATION STUDIES

To deconstruct our CoFiCot framework and validate the contribution of its key components, we conduct comprehensive ablation studies on the overall Coarse-to-fine architecture, classification metrics, and other core design choices (additional experiments are detailed in Appendix D).

**Impact of the Coarse and Fine Stages** We first conduct an ablation to demonstrate the fundamental value of our two stage pipeline. As shown in Table 5, we compare our full model against w/o Coarse Stage and w/o Fine Stage. The w/o Coarse Stage variant sees a notable drop in accuracy, particularly on GSM8K (3.3% ↓). This is consistent with our motivation that indiscriminate refinement leads to over correction on simpler problems, harming overall performance. More significantly, the w/o Fine Stage variant suffers a performance collapse in the difficult MATH dataset (6.7% ↓). This result powerfully demonstrates that our Fine Stage is essential for solving complex problems.

**Analysis of Coarse Classification Metrics** In this part, we analyze the contribution of three individual classification metrics in coarse stage. As detailed in Table 6, we evaluate the performance of the framework when ablating these metrics. The results clearly show that all three metrics contribute positively to the final performance. Any variant using only one or two metrics underperforms the full model across all datasets. The combination of Relia. and Comp. (row 3) performs well, but adding the Cofi. metric (row 5) provides a final boost, particularly on GSM8K (89.7% to 91.8%) and MMLU (67.0% to 68.8%). This confirms that our chosen metrics are not redundant and create a robust and accurate difficulty classifier.

## 5 CONCLUSION

In this paper, we presented CoFiCot, a Coarse-to-fine adaptive framework designed to resolve the efficiency paradox in LLM reasoning. By triaging problems via multi-metric assessment and employing a differentiated refinement strategy, CoFiCot mitigates both over-correction on simple tasks and insufficient refinement on complex ones. The core sequential correction mechanism leverages PRMs to ensure logically coherent repairs by initiating new decoding branches from localized error points. The framework's modularity allows it to seamlessly integrate various reward models, with performance scaling directly alongside the quality of these components. Furthermore, our dynamic early exit protocol ensures computational resources are preserved once reasoning stabilizes, preventing redundant iterations . Empirical results across seven benchmarks demonstrate that CoFiCot significantly outperforms uniform scaling baselines, achieving a 4.0% average accuracy gain on Llama-3-8B while maintaining superior token efficiency. These findings establish CoFiCot as a robust solution for achieving a superior trade-off between accuracy and computational cost. Future work will focus on automating parameter calibration and extending this paradigm to broader domains.

## 6 LIMITATION

The effectiveness of our pipeline depends on the external models guiding refinement and selection processes. Additionally, we acknowledge that for extremely simple queries, generating $k$ samples introduces additional computational overhead compared to greedy decoding. Future work could employ a progressive sampling strategy.

## ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants 62301559, in part by the Natural Science Basis Research Plan in Shaanxi Province of China under Grant No. 2025JC-JCQN-091 and the Open Project Program of State Key Laboratory of Integrated Services Networks of Xidian University under Grant. ISN25-25.

### References

Linjiang Cao, Maonan Wang, and Xi Xiong. A large language model-enhanced q-learning for capacitated vehicle routing problem with time windows. *arXiv preprint arXiv:2505.06178*, 2025.

Justin Chih-Yao Chen, Archiki Prasad, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. Magicore: Multi-agent, iterative, coarse-to-fine refinement for reasoning. *arXiv preprint arXiv:2409.12147*, 2024a.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024b.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthink: Slowdown attacks on reasoning llms. *arXiv preprint arXiv:2502.02542*, 2025.

Loka Li, Zhenhao Chen, Guangyi Chen, Yixuan Zhang, Yusheng Su, Eric Xing, and Kun Zhang. Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. *arXiv preprint arXiv:2402.12563*, 2024.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Dancheng Liu, Amir Nassereldine, Ziming Yang, Chenhui Xu, Yuting Hu, Jiajie Li, Utkarsh Kumar, Changjae Lee, Ruiyang Qin, Yiyu Shi, et al. Large language models have intrinsic self-correction ability. *arXiv preprint arXiv:2406.15673*, 2024.

Li-Chun Lu, Shou-Jen Chen, Tsung-Min Pai, Chan-Hung Yu, Hung-yi Lee, and Shao-Hua Sun. Llm discussion: Enhancing the creativity of large language models via discussion framework and role-play. *arXiv preprint arXiv:2405.06373*, 2024.

Yiran Ma, Zui Chen, Tianqiao Liu, Mi Tian, Zhuo Liu, Zitao Liu, and Weiqi Luo. What are step-level reward models rewarding? counterintuitive findings from mcts-boosted mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24812–24820, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2(4), 2022.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. The right tool for the job: Matching model and instance complexities. *arXiv preprint arXiv:2004.07453*, 2020.

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.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

Yiding Sun, Jihua Zhu, Haozhe Cheng, Chaoyi Lu, Zhichuan Yang, Lin Chen, and Yaonan Wang. Align then adapt: Rethinking parameter-efficient transfer learning in 4d perception. *IEEE Trans. Multimedia*, 2026.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. *Advances in Neural Information Processing Systems*, 37:51118–51168, 2024.

Manya Wadhwa, Xinyu Zhao, Junyi Jessy Li, and Greg Durrett. Learning to refine with fine-grained natural language feedback. *arXiv preprint arXiv:2407.02397*, 2024.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023a.

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

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025.

Yingsen Wang, Dongxu Zhang, Yixiao Li, Weihan Jiao, Guibin Wang, Juanjuan Zhao, Yan Qiang, and Keqin Li. Enhancing power grid resilience with blockchain-enabled vehicle-to-vehicle energy trading in renewable energy integration. *IEEE Transactions on Industry Applications*, 60(2):2037–2052, 2023b.

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

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Ning Yang, Mingrui Fan, Wentao Wang, and Haijun Zhang. Decision-making large language model for wireless communication: A comprehensive survey on key techniques. *IEEE Communications Surveys & Tutorials*, 2025a.

Ning Yang, Hai Lin, Yibo Liu, Baoliang Tian, Guoqing Liu, and Haijun Zhang. Token-importance guided direct preference optimization. *arXiv preprint arXiv:2505.19653*, 2025b.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Tianxiang Sun, Cheng Chang, Qinyuan Cheng, Ding Wang, Xiaofeng Mou, et al. Aggregation of reasoning: A hierarchical framework for enhancing answer selection in large language models. *arXiv preprint arXiv:2405.12939*, 2024.

Dongxu Zhang, Ning Yang, Jihua Zhu, Jinnan Yang, Miao Xin, and Baoliang Tian. Ascot: An adaptive self-correction chain-of-thought method for late-stage fragility in llms. *arXiv preprint arXiv:2508.05282*, 2025.

Dongxu Zhang, Yiding Sun, Pengcheng Li, Yumou Liu, Hongqiang Lin, Haoran Xu, Xiaoxuan Mu, Liang Lin, Wenbiao Yan, Ning Yang, et al. Pointcot: A multi-modal benchmark for explicit 3d geometric reasoning. *arXiv preprint arXiv:2602.23945*, 2026a.

Dongxu Zhang, Yiding Sun, Cheng Tan, Wenbiao Yan, Ning Yang, Jihua Zhu, and Hiajun Zhang. Chain-of-thought compression should not be blind: V-skip for efficient multimodal reasoning via dual-path anchoring. *arXiv preprint arXiv:2601.13879*, 2026b.

Dongxu Zhang, Jihua Zhu, Shiqi Li, Wenbiao Yan, Haoran Xu, Peilin Fan, and Huimin Lu. Igasa: Integrated geometry-aware and skip-attention modules for enhanced point cloud registration. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026c.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

Peijun Zhu, Ning Yang, Jiayu Wei, Jinghang Wu, and Haijun Zhang. Breaking the moe llm trilemma: Dynamic expert clustering with structured compression. *arXiv preprint arXiv:2510.02345*, 2025.

# Supplementary Material

**Dongxu Zhang**[1,2,*] **Hongqiang Lin**[3,*] **Yiding Sun**[1,2,*] **Pengyu Wang**[4]

**Qirui Wang**[1] **Ning Yang**[5,†] **Jihua Zhu**[1,2,†]

[1]School of Software Engineering, Xi'an Jiaotong University

[2]State Key Laboratory of Integrated Services Networks, Xidian University [3]Zhejiang University

[4]The Chinese University of Hong Kong (Shenzhen) [5]Institute of Automation, Chinese Academy of Sciences

## A ALGORITHM

Algorithm 1 formally outlines the execution flow of the CoFiCot framework. The process initiates with Stage 0, where the function `GenerateSolutions` employs stochastic sampling to produce a diverse initial ensemble $\mathcal{R}_0$. This diversity is crucial for the subsequent Stage 1, where `ClassifyDifficulty` aggregates the multi metric signals (entropy, consensus quality, and predicted steps) to assign a difficulty label $D_{\text{final}}$.

The core logic resides in Stage 2, which implements our differentiated routing strategy. For problems identified as *Easy*, the algorithm bypasses the refinement loop entirely, directly proceeding to final aggregation to minimize computational overhead. Conversely, *Medium* and *Hard* problems enter the iterative refinement loop. Within this loop, two key functions drive the optimization.

`ContextAwareCorrection` applies the PRM to localize error steps and regenerates them conditioned on the history of prior corrections, ensuring logical coherence. `SelectTopK` utilizes the ORM to filter the expanded candidate pool $\mathcal{P}$, retaining the highest quality solutions for the next iteration.

Finally, the loop incorporates a dynamic early exit mechanism, which triggers if the reevaluated difficulty $D_{\text{new}}$ converges to *Easy*, preventing redundant iterations before the final `WeightedVoting` aggregation.

## B DETAILED IMPLEMENTATION OF COFICOT

**Confidence Score Transformation** The Shannon entropy $H(\mathcal{A})$ derived in Eq. (2) is mapped to a normalized confidence score $C \in [0, 1]$ using a scaled sigmoid function:

$$C = \sigma(\alpha \cdot (1 - H)), \tag{13}$$

where $\alpha$ is a scaling hyperparameter set to 2.

**Complexity Prediction Prompt** To efficiently estimate the computational complexity $N_{\text{steps}}$ without incurring the cost of full chain generation, we employ a specialized metacognitive prompt $P_{\text{predict}}$. This prompt instructs the model to perform a "look ahead" analysis of the logical depth. The full prompt is presented in Table 7.

**Hyperparameter Justification and Sensitivity** A key design principle of CoFiCot is to remain training free. The hyperparameters used in Stage 1 are chosen based on statistical robustness rather than dataset specific tuning.

**Complexity Thresholds ($d_3$)** The thresholds for *Easy* and *Hard* complexity are not arbitrary integers. They are dynamically derived from the 33rd and 67th percentiles (tertiles) of the solution length distribution of the base model. This ensures that the classifier automatically adapts to different models (e.g., Llama-3 vs. GPT) or prompt styles without manual recalibration.

**Algorithm 1** Coarse-to-fine Adaptive Reasoning

**Input:** Question $Q$; Sampling count $k$; Reasoning path set $\mathcal{R}$; Difficulty label $D$
**Output:** Predicted answer $\hat{A}$

*// Stage 0: Generate initial diverse solution set*
$\mathcal{R}_0 \leftarrow \text{GenerateSolutions}(Q, k)$
*// Stage 1: Multi metric difficulty assessment*
$D_{\text{final}} \leftarrow \text{ClassifyDifficulty}(\mathcal{R}_0, Q)$
*// Stage 2: Difficulty aware routing and refinement*
**if** $D_{\text{final}} =$ *Easy* **then**
  $\mathcal{R}_{\text{final}} \leftarrow \mathcal{R}_0$  *// Skip refinement for efficiency*
**else**
  $\mathcal{R}_{\text{curr}} \leftarrow \mathcal{R}_0$
  $Iter_{\text{max}} \leftarrow \text{SetMaxIter}(D_{\text{final}})$
  **for** $t = 1 \rightarrow Iter_{\text{max}}$ **do**
    *// Step-level correction using PRM feedback*
    $\mathcal{R}_{\text{refined}} \leftarrow \text{ContextAwareCorrection}(\mathcal{R}_{\text{curr}})$
    *// Select top-k solutions via ORM guidance*
    $\mathcal{P} \leftarrow \mathcal{R}_{\text{curr}} \cup \mathcal{R}_{\text{refined}}$
    $\mathcal{R}_{\text{curr}} \leftarrow \text{SelectTopK}(\mathcal{P}, k)$
    *// Check for early convergence*
    $D_{\text{new}} \leftarrow \text{ClassifyDifficulty}(\mathcal{R}_{\text{curr}}, Q)$
    **if** $D_{\text{new}} =$ *Easy* **then**
      **break**
    **end if**
  **end for**
  $\mathcal{R}_{\text{final}} \leftarrow \mathcal{R}_{\text{curr}}$
**end if**
*// Final aggregation via Weighted Self-Consistency*
$\hat{A} \leftarrow \text{WeightedVoting}(\mathcal{R}_{\text{final}})$
**return** $\hat{A}$

**Classification Weights ($w_i$)** We utilize uniform weights ($w_i = 1/3$) for the three metrics. Sensitivity analysis shows that varying these weights (e.g., increasing reliance on Complexity vs. Confidence) results in less than $\pm 0.5\%$ accuracy variance on the MATH dataset. This suggests that the ensemble of metrics is robust, and uniform weighting prevents overfitting to specific distributions.

**Entropy Scaling ($\alpha = 2$)** This scaling factor is selected to map the typical entropy range of LLM outputs to a $[0, 1]$ interval efficiently. Empirical tests indicate that values in the range $\alpha \in [1.5, 2.5]$ yield statistically similar classification boundaries.

**Detailed Mapping of Difficulty Metrics** To facilitate the quantitative synthesis of the multi-metric assessments in Stage 1, the qualitative labels {*Easy, Medium, Hard*} are transformed into numerical ranks. The mapping function $\text{val}(\cdot)$ is defined as:

$$\text{val}(d_i) = \begin{cases} 1 & \text{if } d_i = \text{Easy} \\ 2 & \text{if } d_i = \text{Medium} \\ 3 & \text{if } d_i = \text{Hard} \end{cases}. \tag{14}$$

The final synthesized score $D_{\text{score}}$ is the weighted average of these ranks. With $w_i = 1/3$, the score range $[1, 3]$ is partitioned to decide the final routing path $D_{\text{final}}$.

**Stateful Correction Function $\Phi$** We define the transformation function $\Phi$ as a context-aware re-decoding operator. Unlike standard self-correction which treats the entire trajectory as a flat string, $\Phi$ operates on the reasoning chain as a structured sequence of logical states. $\Phi$ is a mapping that takes the current flawed state of the reasoning chain and produces a rectified successor state. Its internal mechanism can be decomposed into three primary operations. It first constructs a semi-permeable boundary at the temporal index $j$. The verified history $H^{(t)}_{i,j-1}$ is treated as a fixed prefix, ensuring that the correction does not violate the established logical premises. The function $\Phi$ transforms the raw PRM signal $\mathcal{F}_{i,j}$ into a hidden instructional

**Complexity Prediction Prompt ($P_{\text{predict}}$)**

**System Instruction:**
You are an expert Logic Complexity Evaluator. Your task is to estimate the reasoning depth required to solve a problem without generating the full solution.
**Definition of a Step:**
A step is defined as a distinct atomic reasoning operation, such as:

- Extracting a specific variable or condition.
- Performing a single arithmetic calculation.
- Making a logical deduction or transition.

**Input Problem:**
{Input Question $Q$}
**Task:**
Analyze the problem structure and predict the minimum number of steps required to the correct solution.
**Output Constraint:**
Do NOT output the reasoning process or the answer. Output **ONLY** a single integer representing the estimated step count (e.g., 5).

Table 7: The prompt used for estimating $N_{\text{steps}}$.

| Criterion | GSM8K | MATH |
|---|---|---|
| Pessimistic | 91.4 | 47.2 |
| Optimistic | 90.8 | 46.5 |
| Democratic | 90.3 | 45.8 |
| CoFiCot(ours) | **91.8** | **47.9** |

Table 8: Ablation study on the criterion for detecting problem difficulty. We compare our method against three alternative strategies: Pessimistic, Optimistic, and Democratic. We report accuracy(%).

vector. The output of $\Phi$ is not merely a single replaced step $s_{i,j}^{(t)}$, but a new initial state for the subsequent autoregressive decoding. The operator triggers a recursive generation:

$$\mathcal{R}_t = H_{i,j-1}^{(t)} \cup \{s_{i,j}^{(t)}\} \cup \text{Gen}\left(Q, H_{i,j-1}^{(t)}, s_{i,j}^{(t)}\right), \tag{15}$$

where $\text{Gen}(\cdot)$ represents the model's completion of the remaining trajectory.

## C ANALYSIS

**Computational Overhead and Latency Analysis** A concern regarding the lightweight nature of Stage 0 is the generation of $k = 40$ samples compared to a simple greedy decoding ($k = 1$) for trivial queries (e.g., $1 + 1 =?$). We distinguish here between Total Compute (FLOPs) and User Perceived Latency. To Latency Profile. The generation of the initial ensemble $\mathcal{R}_0$ is fully parallelizable. On modern GPU clusters, the time complexity of generating $k = 40$ sequences is $O(1)$ relative to the batch size (assuming sufficient VRAM), comparable to generating a single sequence. In contrast, heavy reasoning models (like OpenAI o1) or iterative refinement methods operate sequentially ($O(N)$). Therefore, for *Easy* problems that bypass Stage 2, CoFiCot maintains a low latency profile competitive with standard parallel sampling.

To Comparison with Greedy Decoding. While $k = 40$ consumes more energy than $k = 1$ greedy decoding, strictly using $k = 1$ risks hallucination on deceptively simple questions. Our ablation (refer to Confidence Assessment in Table 6) shows that the ensemble based consistency check is crucial for filtering out these confident errors. For resource constrained environments, a progressive sampling strategy can be employed. Starting with a small $k$ (e.g., $k = 5$) and only scaling to $k = 40$ if the initial confidence score falls below a safety threshold.

| Modules | MMLU | MATH |
|---|---|---|
| ORM-Only | 67.8 | 47.1 |
| PRM-Only | 67.3 | 46.5 |
| Both | **68.8** | **47.9** |

Table 9: Ablation study on the final answer selection, using ORM-only, PRM-only or both. We also report accuracy(%).

# D ABLATION EXPERIMENTS

**Analysis of Difficulty Aggregation Strategy** In coarse stage, after obtaining three distinct difficulty labels ($d_1, d_2, d_3$), we synthesize them using a balanced averaging strategy. In Table 8, we compare this method to three other common methods: Pessimistic which classifies a problem as hard if any metric flags it as such; Optimistic which requires all metrics to agree; and Democratic. The results show that our CoFiCot balanced averaging strategy achieves the best performance on both GSM8K (91.8%) and MATH (47.9%). The Pessimistic strategy is a close second, suggesting that it is generally better to err on the side of caution and refine more often. Conversely, the Democratic and Optimistic strategies perform worst, as they are more likely to misclassify a hard problem as easy, causing it to skip the crucial refinement stage.

**Synergy of PRM and ORM in Refinement** Finally, we analyze the synergistic roles of the PRM and ORM within our Fine Stage. In our design, the PRM provides granular, step-level feedback for correction, while the ORM provides holistic, solution level scores for selection. In Table 9, we ablate this design. In the ORM-Only variant, we remove the PRM-guided feedback and rely on LLM self feedback for correction, while still using the ORM for selection. In the PRM-Only variant, we use the PRM for correction but also use its aggregated step scores for selection, removing the ORM. The results demonstrate that both RMs are essential. The ORM-Only variant suffers, indicating that unguided self feedback is inferior to the PRM's precise error localization. The PRM-Only variant also performs worse, suggesting that a step-level score aggregator is a poor proxy for the holistic quality of a complete solution, which the ORM is trained to assess. The full model achieves the highest accuracy, confirming the necessity of our dual RM design where each model performs its specialized function.

# E REFINEMENT IMPLEMENTATION DETAILS

**Handling Easy Problems** For problems classified as *Easy*, we assume the initial ensemble $\mathcal{R}_0$ contains sufficient signal. We apply Weighted Self-Consistency [Li et al., 2023], where the final answer is selected by voting, with each vote weighted by the solution's ORM score. This avoids the computational overhead of the iterative loop while ensuring robust aggregation.

**Dynamic Termination Criteria** To prevent redundant computation, the iterative loop in Stage 2 employs a dynamic stopping mechanism. At the end of each iteration $t$, the new solution set $\mathcal{R}_t$ is reevaluated using the Stage 1 classifier. The loop terminates immediately if: The difficulty of $\mathcal{R}_t$ is reclassified as *Easy*, indicating that the solutions have stabilized into a high confidence, high consensus state. The process reaches the predefined maximum iteration count ($Iter_{\text{max}}$), set to 2 for *Medium* and 3 for *Hard* problems.

# F MODULARITY AND GENERALIZATION

**Dependency on Reward Models** The CoFiCot framework is designed to be modular. While our primary experiments on mathematical reasoning (MATH, GSM8K) utilize the specialized Math-Shepherd-7B as a PRM, the framework does not strictly require a domain specific PRM to function.

**Implementation of General Reasoning** For domains like Commonsense Reasoning (ARC) where dense step-level supervision data (and thus high quality specialized PRM) are scarce, CoFiCot adapts its configuration: Instead of a trained PRM, we employ a general purpose LLM prompted as a verifier to provide step-level feedback. As shown in our ablation study (Table 9), the ORM alone contributes significantly to performance. In the absence of a strong PRM, the framework relies more heavily on the ORM for trajectory selection in Stage 2. This flexibility allows CoFiCot to generalize to Date Understanding and "ARC" tasks (Table 2), demonstrating that the benefit of sequential correction, maintaining logical coherence during repair persists even with general purpose verifiers.