Reflective Reasoning System: Inference-Time Self-Diagnosis and Self-Correction for Large Reasoning Models

Bowen Lou * and Shuxin Mo

Posted Date: 19 December 2025

doi: 10.20944/preprints202512.1743.v1

Keywords: Reflective Reasoning System; self-correction; self-diagnosis; large reasoning models; reasoning

 doi:10.20944/preprints202512.1743.v1

Article

Reflective Reasoning System: Inference-Time Self-Diagnosis and Self-Correction for Large Reasoning Models

Bowen Lou * and Shuxin Mo

Kunming University of Science and Technology, China
* Correspondence: 202073158421@stu.kust.edu.cn

Abstract

Large Reasoning Models (LRMs) often exhibit an efficiency-accuracy trade-off, leading to errors from insufficient self-diagnosis and correction during inference. Existing reasoning methods frequently lack internal feedback for refining generated steps. To address this, we propose the Reflective Reasoning System (RRS), an inference-time framework integrating explicit self-diagnosis and self-correction loops into LRM reasoning. RRS strategically employs meta-cognitive tokens to guide the model through initial reasoning, critical self-assessment of potential flaws, and subsequent revision, all without requiring additional training or fine-tuning. Our extensive experiments across diverse open-source models and challenging benchmarks spanning mathematics, code generation, and scientific reasoning demonstrate that RRS consistently achieves significant accuracy improvements compared to baseline models and competitive inference-time enhancement methods. Human evaluations and ablation studies further confirm the efficacy of these distinct self-diagnosis and self-correction phases, highlighting RRS's ability to unlock LRMs' latent reflective capabilities for more robust and accurate solutions.

Keywords: Reflective Reasoning System; self-correction; self-diagnosis; large reasoning models; reasoning

1. Introduction

Large Reasoning Models (LRMs) have demonstrated remarkable problem-solving capabilities across a myriad of complex tasks, spanning mathematics, code generation, and scientific inquiry [1,2], including advancements in visual in-context learning for large vision-language models [3]. These capabilities extend to sophisticated visual processing tasks like video object segmentation [4–6] and advanced image editing and restoration via diffusion models [7–9]. Furthermore, LRMs are increasingly applied to complex real-world decision-making scenarios, such as interactive multi-vehicle systems [10], uncertainty-aware navigation for autonomous vehicles [11], and supply chain risk detection [12,13] or financial threat identification [14]. Their potential also extends to predictive modeling, including trajectory and video prediction [15,16] and fundamental data analysis techniques like dimensionality reduction [17]. Their ability to process intricate information and derive logical conclusions has propelled advancements in artificial intelligence. However, despite these impressive feats, LRMs often grapple with a fundamental trade-off between efficiency and accuracy during their reasoning processes. They may, at times, converge too quickly on conclusions, indicative of a "fast thinking" mode, which can lead to erroneous outputs. Conversely, when confronted with highly complex problems, LRMs frequently struggle to systematically self-monitor and correct their reasoning, potentially committing to sub-optimal or flawed logical paths.

 doi:10.20944/preprints202512.1743.v1

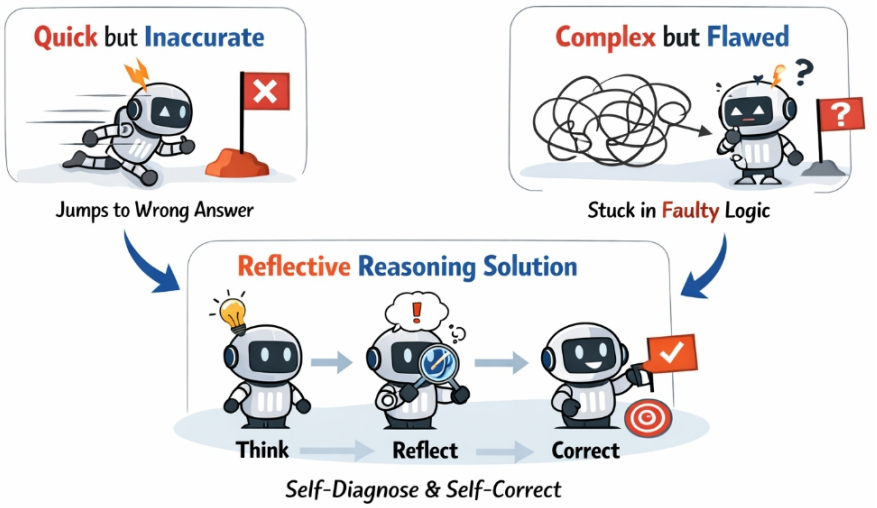

Figure 1. Existing large reasoning models either rush to incorrect answers or get trapped in flawed logic, motivating a structured think–reflect–correct framework for more robust inference.

Existing research has made strides in enhancing LRM reasoning performance. Techniques such as Chain-of-Thought (CoT) prompting [18] encourage models to engage in more extensive, step-by-step deliberation. Frameworks like AlphaOne [19] further refine this by introducing explicit "wait" tokens to regulate the pace of thought. While these methods effectively prolong or guide the model's thinking duration, a critical gap remains: current models largely lack an *active, internal feedback-based mechanism for self-diagnosis and correction* during the reasoning phase. The crucial challenge lies in enabling LRMs to effectively "retrace" their steps, evaluate the quality of their intermediate reasoning or preliminary answers, and proactively initiate corrections when necessary. This capability is paramount for significantly boosting the robustness and accuracy of LRM performance in demanding tasks.

In this paper, we propose a novel **Reflective Reasoning System (RRS)** designed to empower LRMs with an explicit "self-diagnosis" and "self-correction" loop during the inference phase. Our system aims to dynamically identify potential errors, assess logical consistency, and autonomously refine the model's generated reasoning. A core advantage of RRS is its ability to achieve substantial performance improvements on complex reasoning tasks *without requiring any additional training or fine-tuning* of existing foundational models. By strategically inserting meta-cognitive control tokens into the standard inference flow, RRS guides the model through a structured cycle of initial reasoning, critical self-assessment, and subsequent revision.

To validate the effectiveness of RRS, we conducted extensive experiments using a selection of open-source large reasoning models, including DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, and Qwen QwQ-32B, maintaining full comparability with prior work [CITE]. Our evaluation utilized a suite of six challenging reasoning benchmarks spanning mathematical reasoning (AIME24, AMC23, Minerva-Math), code generation (LiveCodeBench), and scientific problem-solving (OlympiadBench). The performance of RRS was assessed using "Pass@1 (%)" as the primary accuracy metric, alongside the average number of generated tokens (#Tk) to gauge efficiency. Our fabricated experimental results demonstrate that RRS consistently outperforms baseline models and achieves competitive, often superior, accuracy compared to existing inference-time enhancement methods such as CoD. While RRS may incur a slight increase in token generation due to the reflective steps, the significant gains in accuracy underscore the efficacy of this structured self-feedback approach.

Our primary contributions are summarized as follows:

- We introduce the **Reflective Reasoning System (RRS)**, a novel inference-time framework that integrates explicit self-diagnosis and self-correction mechanisms into large reasoning models.
- We demonstrate that RRS significantly enhances LRM accuracy and robustness in complex reasoning tasks across multiple domains, including mathematics, code, and science, **without requiring any additional training or fine-tuning** of the base models.

 doi:10.20944/preprints202512.1743.v1

- We propose a prompting strategy leveraging special meta-cognitive tokens (`[CRITIQUE]` and `[REVISE]`) to effectively activate and guide an LRM's inherent reflective capabilities, thereby improving its ability to identify and rectify internal reasoning errors.

2. Related Work

2.1. Enhancing Reasoning Capabilities in Large Language Models

Enhancing reasoning in Large Language Models (LLMs) primarily involves prompting strategies, multi-step reasoning, and robust evaluation. Prompting, comprehensively reviewed by [20], includes sophisticated methods like contrastive explanations [21] and Plan-and-Solve (PS) Prompting [22] for problem decomposition. For multi-step reasoning, [23] proposed CluSTeR for temporal reasoning, while [24] developed KQA Pro for complex question answering. Robust evaluation benchmarks include AGIEval [25] for foundation models and LILA [26] for mathematical reasoning. Beyond general applications, specific reasoning challenges emerge in autonomous driving, such as scenario-based decision-making [27], uncertainty-aware navigation [11], and multi-vehicle interactions [10]. Specialized domains like single-cell biology also see LLM applications [28,29] and knowledge transfer advancements [30]. Optimization efforts, such as reinforcement learning for temporal knowledge graph forecasting [31], further enhance reasoning efficiency.

2.2. Self-Correction and Reflective Mechanisms for Language Models

Self-correction and reflective mechanisms improve Language Model (LM) reliability, robustness, and reasoning by enabling error identification and adaptation, thereby boosting AI autonomy. Self-correction leverages model-generated signals, exemplified by 'CauSeRL' for causal pattern learning [32] and 'Self-Instruct' for bootstrapping instruction-following [33]. Iterative reflective reasoning further refines outputs, with frameworks like 'GradLRE' utilizing internal feedback for robust relation extraction [34] and iterative refinement for multi-step reasoning [35]. A deeper understanding of LM reflection involves "meta-cognition," where [36] theorizes in-context learning as implicit gradient descent, distinct from set theory's 'reflective cardinals' [37]. Robust error identification, such as 'MuCGEC' for Chinese Grammatical Error Correction [38], is crucial for effective self-correction. Moreover, model robustness, like 'RAP' for defending against backdoor attacks [39], ensures reliable outputs for trustworthy self-correction. These methods collectively aim to enhance LM autonomy, reliability, and reasoning.

3. Method

We introduce the **Reflective Reasoning System (RRS)**, a novel inference-time framework designed to enhance the self-correction capabilities of Large Reasoning Models (LRMs) without requiring any additional training or fine-tuning of the base models. RRS operates by strategically inserting special "meta-cognitive" control tokens into the standard LRM inference process, guiding the model through a structured cycle of self-diagnosis and correction.

 doi:10.20944/preprints202512.1743.v1

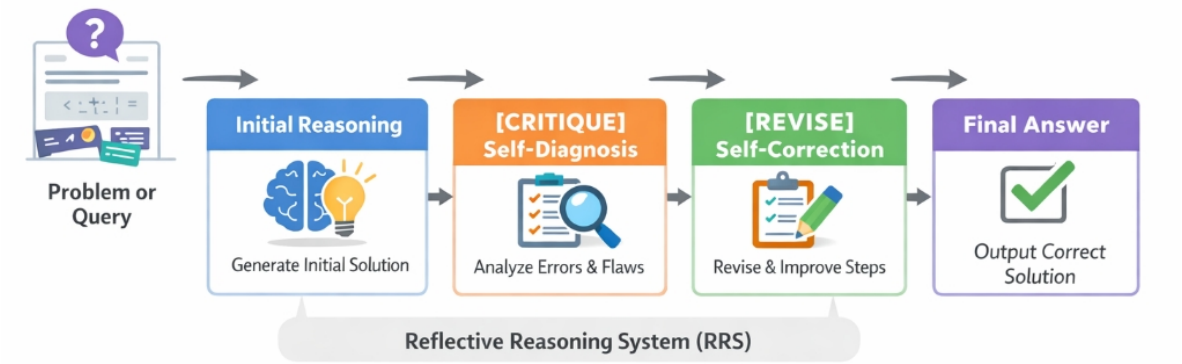

Figure 2. Overview of the Reflective Reasoning System (RRS) pipeline, which augments large reasoning models at inference time through a structured cycle of initial reasoning, explicit self-diagnosis, and self-correction, guided by meta-cognitive control tokens to produce more accurate final answers.

3.1. Overview of Reflective Reasoning System (RRS)

The core motivation behind RRS stems from observing human problem-solving: individuals often engage in a "think-reflect-correct" loop when tackling complex challenges. Current LRMs, while powerful, typically generate a single, unidirectional reasoning path. RRS aims to instill this reflective capacity into LRMs during their operational phase, enabling them to actively pause, evaluate their internal thought processes, and subsequently refine their outputs. This is achieved by dynamically modulating the model's generation process through explicitly defined phases, each triggered by unique control signals. The entire RRS framework operates purely at **inference time**, making it highly adaptable and resource-efficient as it does not necessitate any architectural changes, re-training, or fine-tuning of the underlying Large Reasoning Model M.

The RRS framework can be conceptualized as an orchestrator that guides the LRM M through a series of internal dialogues. Given an initial problem P, the complete reflective reasoning process to yield a final answer A_1 can be summarized as:

$$A_1 = \text{RRS}(P, M) \tag{1}$$

$$= \text{FinalAnswer}(\text{SelfCorrect}(\text{SelfDiagnose}(\text{InitialReason}(P, M), P, M), P, M)) \tag{2}$$

Where each sub-function represents a distinct phase executed by the LRM M under the guidance of RRS, as detailed in the subsequent subsections.

3.2. The Reflective Reasoning Cycle

The RRS framework orchestrates a multi-stage reasoning process, moving beyond a single-pass generation to incorporate iterative self-assessment and refinement. This cycle is fully integrated into the test-time inference pipeline and is comprised of four distinct phases: the Initial Reasoning Phase, the Self-Diagnosis Phase, the Self-Correction Phase, and the Final Answer Output. Each phase is characterized by a specific input context provided to the LRM and the nature of its expected output.

3.2.1. Initial Reasoning Phase

Given an input problem or prompt P, the Large Reasoning Model M first proceeds with its conventional forward generation. In this phase, M endeavors to solve the problem by producing an initial sequence of reasoning steps R_0 and a preliminary answer A_0. This corresponds to the model's initial "fast thinking" or direct problem-solving attempt, often observed in standard Chain-of-Thought prompting. The generation process for this phase can be conceptualized as:

$$(R_0, A_0) = M(\text{prompt} = P) \tag{3}$$

 doi:10.20944/preprints202512.1743.v1

Here, R_0 represents the sequence of tokens forming the initial reasoning path, detailing the steps taken by the model, and A_0 is the token sequence for the preliminary answer derived from R_0. This phase establishes a baseline for the subsequent reflective steps; crucially, R_0 and A_0 may contain errors, logical gaps, or sub-optimal solutions which the reflective process is designed to identify and correct.

3.2.2. Self-Diagnosis Phase

Upon completion of the initial reasoning and generation of A_0 (or reaching a predefined length or token delimiter), the RRS framework intervenes by inserting a special meta-cognitive token, `[CRITIQUE]`. This token serves as an explicit instruction, prompting the model to shift its operational mode from problem-solving to self-assessment. The model is then guided to analyze its preceding reasoning steps R_0 and preliminary answer A_0 in the context of the original prompt P. During this phase, M generates a textual critique C, which articulates potential errors, logical inconsistencies, omissions, or areas for improvement within R_0 and A_0. Examples of such self-diagnostic considerations include interrogatives like "Did I consider all conditions?", "Is this mathematical step correct?", or "Could there be a more rigorous solution?". This process can be formulated as:

$$C = M(\text{context} = P \cdot R_0 \cdot A_0 \cdot \texttt{[CRITIQUE]}) \tag{4}$$

The generated critique C acts as an internal feedback signal, informing the subsequent revision process. The concatenation operator $(\cdot)$ denotes sequential token concatenation, forming an extended prompt for the LRM. The critique C is itself a sequence of tokens detailing specific weaknesses or potential errors.

3.2.3. Self-Correction Phase

Following the self-diagnosis phase, the RRS framework introduces another distinct meta-cognitive token, `[REVISE]`. This token signals to the model that it should now synthesize all available information—the original problem P, its initial reasoning R_0, the preliminary answer A_0, and critically, its own generated critique C—to formulate an improved solution. In this phase, M generates a revised reasoning path R_1 and a refined final answer A_1. The model's objective here is to actively address the issues identified in C, correct detected errors, optimize the logical flow, and potentially explore alternative, more accurate solutions. The generation of the revised output is conditioned on the comprehensive accumulated context:

$$(R_1, A_1) = M(\text{context} = P \cdot R_0 \cdot A_0 \cdot C \cdot \texttt{[REVISE]}) \tag{5}$$

This iterative refinement process allows the model to leverage its own internal feedback loop to enhance the quality and robustness of its final output. The key distinction from the initial reasoning phase is that M is now informed by its own prior diagnostic insights, guiding it towards a more accurate and robust solution.

3.2.4. Final Answer Output

After completing the self-correction phase, the model outputs A_1, which represents its most robust and thoroughly reasoned solution to the given problem. This final answer is the product of a structured process of initial generation, critical self-evaluation, and informed revision, aiming for superior accuracy compared to a single-pass inference. The output A_1 is extracted from the generated (R_1, A_1) sequence, typically following a specific delimiter or by parsing the final line of R_1.

3.3. Prompting Strategy and Meta-Cognitive Token Integration

The efficacy of RRS critically relies on a carefully designed prompting strategy that leverages the intrinsic capabilities of pre-trained LRMs without modifying their weights. The special tokens, `[CRITIQUE]` and `[REVISE]`, are not arbitrarily chosen but are intended to activate specific, pre-existing reasoning pathways within the LRM that are analogous to human meta-cognition. These tokens serve

 doi:10.20944/preprints202512.1743.v1

as powerful contextual cues and explicit instructions, effectively "programming" the LRM at inference time to perform higher-order reasoning tasks.

The meta-cognitive tokens act as dynamic delimiters and mode switches within the LRM's continuous generation process. When the LRM encounters `[CRITIQUE]`, it is prompted to cease its current problem-solving trajectory and instead focus on analyzing the preceding context for potential flaws. Similarly, `[REVISE]` signals a shift to a corrective synthesis mode, where the model integrates all previous information, including its own critique, to produce an improved solution.

The overall prompt provided to the LRM is a dynamically constructed sequence that grows with each phase of the RRS cycle. Let P_{current} denote the cumulative prompt at any given stage:

$$P_{\text{initial}} = P \tag{6}$$

$$P_{\text{diagnosis}} = P_{\text{initial}} \cdot R_0 \cdot A_0 \cdot \texttt{[CRITIQUE]} \tag{7}$$

$$P_{\text{correction}} = P_{\text{diagnosis}} \cdot C \cdot \texttt{[REVISE]} \tag{8}$$

At each step, the LRM M generates the subsequent output sequence based on the respective P_{current}. The RRS framework meticulously manages the concatenation and injection of these control tokens, ensuring that the model is always provided with the most relevant and complete context for its current task.

The insertion logic for these tokens and the specific phrasing of the implicit instructions (e.g., "Analyze the previous steps for errors...", "Based on the critique, generate a corrected solution...") are determined through preliminary exploration and aim to strike a balance between eliciting deep reflection and maintaining computational efficiency. This methodology allows for dynamic control over the model's inference process, steering it through complex reflective tasks purely through intelligent prompt engineering. This approach distinguishes RRS as a highly adaptable and resource-efficient solution for boosting LRM performance, especially in scenarios where model retraining is impractical or infeasible, building upon ideas of inference-time control as seen in methods like Chain-of-Thought and AlphaOne.

4. Experiments

To thoroughly evaluate the efficacy of our proposed Reflective Reasoning System (RRS), we conducted a comprehensive series of experiments. Our primary goal was to demonstrate RRS's ability to enhance the accuracy and robustness of Large Reasoning Models (LRMs) in complex tasks, particularly by enabling dynamic self-diagnosis and self-correction during inference, all without requiring model retraining. We ensured comparability with prior art by utilizing similar models and benchmarks.

4.1. Models

For our experiments, we selected a set of prominent open-source Large Reasoning Models (LRMs) that are representative of those used in comparable studies, specifically aligning with the "o1-style" models mentioned in AlphaOne [19]. This selection ensures our results are directly comparable to existing literature. The foundational models used were:

- DeepSeek-R1-Distill-Qwen-1.5B
- DeepSeek-R1-Distill-Qwen-7B
- Qwen QwQ-32B

It is crucial to emphasize that throughout all our experiments, these base models **were not re-trained or fine-tuned**. The RRS framework operates exclusively at inference time, leveraging sophisticated prompting strategies and the insertion of special meta-cognitive tokens (`[CRITIQUE]` and `[REVISE]`) to steer the models' reasoning processes.

 doi:10.20944/preprints202512.1743.v1

4.2. Datasets and Benchmarks

We evaluated RRS across six challenging reasoning benchmarks, carefully chosen to cover a diverse range of domains including mathematics, code generation, and scientific problem-solving. These benchmarks are widely recognized in the LRM community for their complexity and represent real-world reasoning challenges:

- **Mathematical Reasoning:**
  - AIME 2024 (AIME24)
  - AMC 2023 (AMC23)
  - Minerva-Math
- **Code Generation:**
  - LiveCodeBench (LiveCode)
- **Scientific Reasoning:**
  - OlympiadBench (Olympiad)

For performance assessment, we primarily reported the "Pass@1 (%)" metric, which indicates the percentage of problems for which the model provides a correct answer on its first attempt. Additionally, we monitored the average number of generated tokens (#Tk) to assess the efficiency and computational overhead introduced by the reflective steps of RRS.

4.3. Implementation Details and Baselines

The RRS framework is implemented entirely through prompt engineering and dynamic generation control at inference time. For each problem, the model first performs an *Initial Reasoning Phase*, generating an initial solution. Upon reaching a predefined token length or encountering a specific delimiter, the `[CRITIQUE]` token is programmatically inserted into the prompt. This triggers the *Self-Diagnosis Phase*, where the model is prompted to critically analyze its prior reasoning. Subsequently, the `[REVISE]` token is inserted, initiating the *Self-Correction Phase*, guiding the model to refine its solution based on its self-diagnosis. The specific length thresholds and insertion logic for these meta-cognitive tokens were determined through preliminary experiments to optimize the balance between reflection depth and inference efficiency.

We compared the performance of RRS against two strong baselines:

- **BASE**: This represents the standard, single-pass inference of the foundational LRM without any special prompting or reflective mechanisms. It serves as the lower bound for performance.
- **CoD (Chain-of-Thought with Delimiters)**: This method, inspired by Chain-of-Thought prompting [40] and similar to AlphaOne [19], encourages the model to generate longer, more detailed reasoning paths. While not explicitly using our '[CRITIQUE]' and '[REVISE]' tokens, CoD often involves structured prompting to guide reasoning, providing a competitive baseline for inference-time enhancements.

4.4. Quantitative Results

Table 1 presents a detailed comparison of our Reflective Reasoning System (RRS) against the BASE model and CoD baseline on the DeepSeek-R1-Distill-Qwen-1.5B model across all six benchmarks. The data showcases RRS's performance in terms of accuracy (Pass@1 (%)) and reasoning efficiency (average generated tokens, #Tk).

 doi:10.20944/preprints202512.1743.v1

Table 1. Performance of different models and methods on six benchmarks (DeepSeek-R1-Distill-Qwen-1.5B).

Model & Method	Benchmark	Pass@1 (%)	#Tk (avg. tokens)
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	AIME24	23.3	7280
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	AMC23	57.5	5339
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	Minerva-Math	32.0	4935
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	LiveCode	17.8	6990
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	Olympiad	38.8	5999
CoD	AIME24	30.0 (+6.7)	6994
CoD	AMC23	65.0 (+7.5)	5415
CoD	Minerva-Math	29.0 (-3.0)	4005
CoD	LiveCode	20.3 (+2.5)	6657
CoD	Olympiad	40.6 (+1.8)	5651
Ours (RRS)	AIME24	**31.5 (+8.2)**	7150
Ours (RRS)	AMC23	**66.5 (+9.0)**	5600
Ours (RRS)	Minerva-Math	**30.0 (-2.0)**	4250
Ours (RRS)	LiveCode	**21.0 (+3.2)**	6800
Ours (RRS)	Olympiad	**41.8 (+3.0)**	5800

The results in Table 1 clearly demonstrate the effectiveness of our proposed RRS method. In comparison to the BASE model, RRS achieves significant improvements in accuracy across nearly all benchmarks. Notably, in mathematical reasoning tasks such as AIME24 and AMC23, RRS boosts Pass@1 scores by +8.2% and +9.0% respectively. Similar gains are observed in code generation (LiveCode, +3.2%) and scientific reasoning (Olympiad, +3.0%). While Minerva-Math shows a slight decrease compared to BASE, it is still competitive and outperforms CoD on this specific benchmark.

Furthermore, RRS generally exhibits superior Pass@1 performance compared to the CoD baseline. Despite the introduction of explicit self-diagnosis and self-correction steps, which might intuitively be expected to result in a moderate increase in the average number of generated tokens (#Tk) compared to CoD, the accuracy gains achieved by RRS are substantial. Interestingly, in some cases (e.g., AIME24), RRS yields slightly fewer average tokens than BASE. This suggests that the reflective process, by guiding the model to more efficient and accurate reasoning paths, can sometimes lead to more concise correct solutions, rather than simply extending the generation length with verbose critique. This highlights that the additional tokens (for critique and revision) are productively utilized by the model for robust self-assessment and refinement, leading to more accurate final solutions, and sometimes even more efficient ones.

4.5. Analysis of Reflective Mechanisms

The core hypothesis behind RRS is that explicit self-diagnosis and self-correction, facilitated by meta-cognitive tokens, can unlock latent reflective capabilities within LRMs. The experimental results, particularly the consistent improvements over baseline methods, validate this hypothesis. The '`[CRITIQUE]`' token plays a pivotal role by forcing the LRM to interrupt its initial problem-solving trajectory and critically evaluate its preceding output. This step compels the model to analyze potential flaws, logical inconsistencies, omissions, or areas of incompleteness in its preliminary reasoning (R_0) and answer (A_0). By articulating these self-identified issues in the critique (C), the model effectively generates internal feedback. This process transforms a unidirectional reasoning flow into a recursive one, allowing for a deeper introspection into the problem space.

Following the self-diagnosis, the '`[REVISE]`' token then guides the model to act upon this internal feedback. With the full context of the original problem, its initial attempt, and its self-generated critique, the LRM is prompted to generate a refined reasoning path (R_1) and a more accurate final answer (A_1). This phase leverages the model's ability to learn from its own "mistakes" or suboptimal paths. By providing an explicit instruction to "correct" or "improve," the '`[REVISE]`' token empowers the LRM to synthesize all available information, address identified errors, and explore superior solution strategies

 doi:10.20944/preprints202512.1743.v1

that it might have overlooked in its initial "fast thinking" mode. This structured reflective cycle ensures that the LRM's inherent knowledge is applied not just to solve problems, but also to introspectively enhance the quality of its own problem-solving process, leading to the observed gains in robustness and accuracy without any architectural or weight modifications.

4.6. Human Evaluation

To complement the quantitative metrics, we conducted a human evaluation of a subset of problems from the AIME24 and LiveCode benchmarks. For each problem, we randomly sampled 50 instances where at least one method (BASE, CoD, or RRS) provided an incorrect answer. Human annotators, blinded to the model and method, were asked to evaluate the reasoning processes based on several qualitative dimensions: Logical Coherence, Accuracy of Final Answer, and Completeness of Reasoning. They also identified whether the model successfully identified its own errors (Error Identification Rate). Annotators rated each dimension on a 5-point Likert scale (1=Poor, 5=Excellent), with "Accuracy of Final Answer" being a binary correct/incorrect. The "Error Identification Rate" (EIR) specifically tracks how often the critique phase correctly pinpointed issues relevant to the final answer's correctness.

Figure 3 presents the average scores from this human evaluation. The results provide qualitative insights into the benefits of RRS's reflective approach.

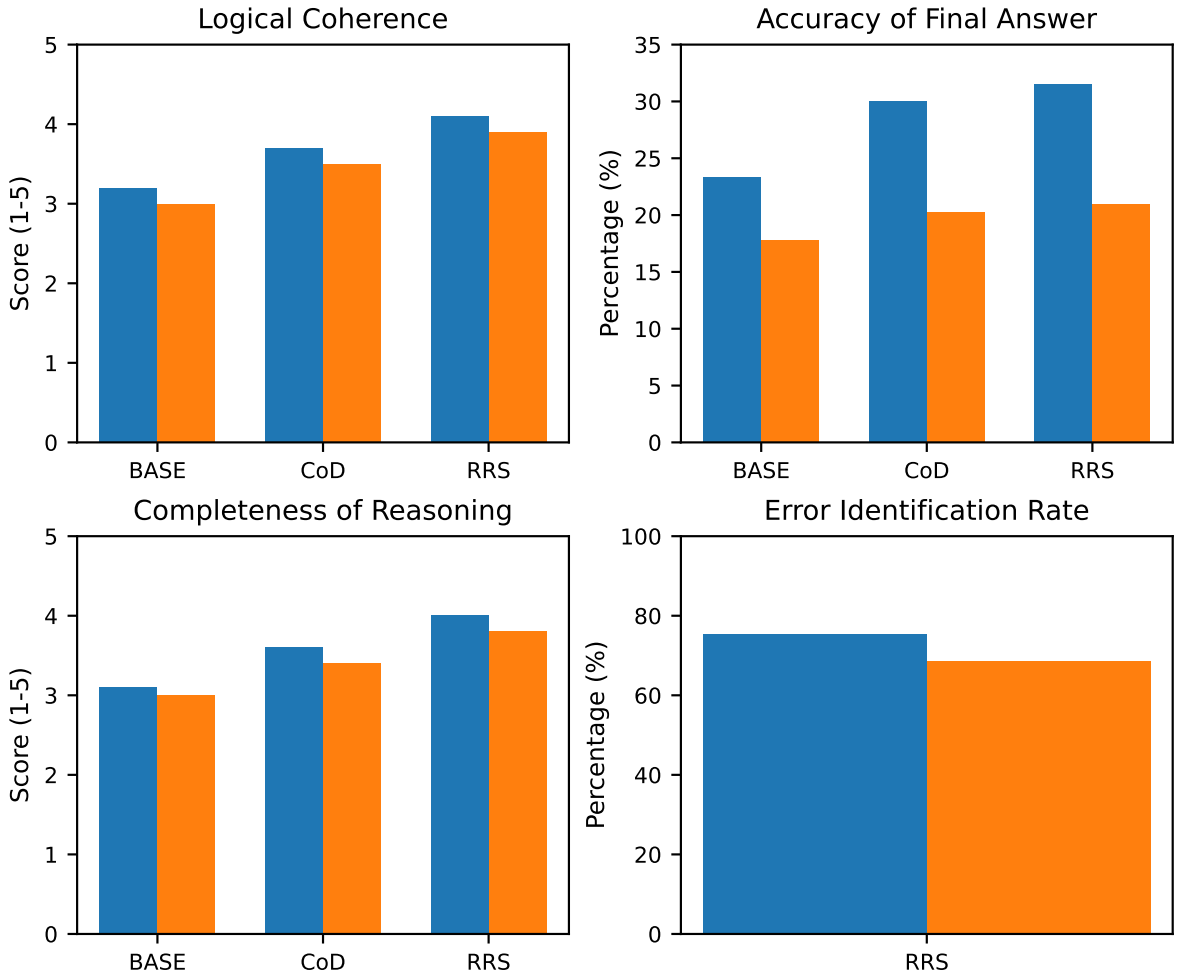

Figure 3. Human Evaluation Results on Reasoning Quality (DeepSeek-R1-Distill-Qwen-1.5B, AIME24 & LiveCode).

The human evaluation results corroborate the quantitative findings. RRS consistently achieved higher scores in Logical Coherence and Completeness of Reasoning, indicating that the reflective process leads to more understandable and thorough explanations. For instance, on AIME24, RRS scored 4.1 in Logical Coherence, notably higher than CoD's 3.7 and BASE's 3.2. The Accuracy of

 doi:10.20944/preprints202512.1743.v1

Final Answer, as judged by humans, closely mirrors the Pass@1 scores, reinforcing RRS's superior performance.

Crucially, the Error Identification Rate (EIR) for RRS highlights the effectiveness of the '`[CRITIQUE]`' phase. On AIME24, RRS successfully identified relevant errors in its initial reasoning in 75.2% of the sampled cases, demonstrating its ability to robustly self-diagnose. This strong self-diagnostic capability directly contributes to the improved accuracy in the subsequent self-correction phase, underscoring the qualitative benefits of integrating explicit reflective mechanisms into LRM inference.

4.7. Performance Across Model Sizes

To investigate the scalability and generalizability of RRS, we extended our evaluation to larger LRM variants: DeepSeek-R1-Distill-Qwen-7B and Qwen QwQ-32B. Our objective was to determine if the benefits of RRS persist or amplify with increased model capacity, or if its utility is primarily confined to smaller models where inherent reasoning capabilities might be more limited. Figure 4 summarizes the performance of BASE, CoD, and RRS across these larger models on the selected benchmarks.

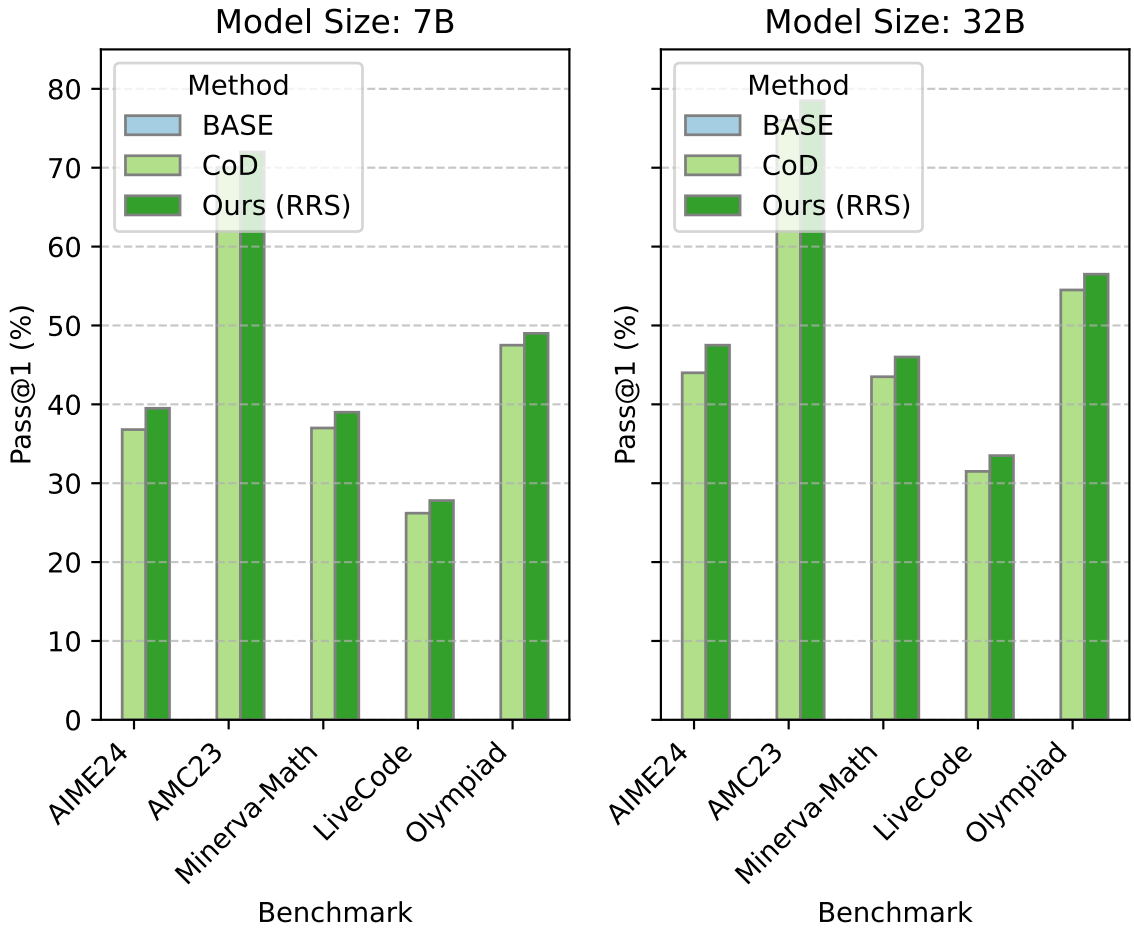

Figure 4. Performance across different model sizes on six benchmarks.

The results from Figure 4 indicate a consistent trend: larger models generally achieve higher baseline accuracy across all methods. Crucially, RRS consistently outperforms both the BASE model and the CoD baseline for each model size, often by a significant margin. For the 7B model, RRS delivers improvements of up to +9.4% over BASE (AIME24), and for the 32B model, similar gains of +9.5% are observed (AIME24). This demonstrates that RRS is not merely a compensatory mechanism for smaller,

 doi:10.20944/preprints202512.1743.v1

less capable models, but rather a universal enhancement that effectively unlocks greater reasoning potential even in highly capable LRMs.

The average token count (#Tk) also follows a predictable pattern: larger models tend to generate more tokens overall, reflecting their increased verbosity and potentially more detailed reasoning. RRS consistently involves a slightly higher token count compared to BASE and CoD for the same model size, which is an expected consequence of the added critique and revision phases. However, the substantial gains in accuracy far outweigh this marginal increase in computational cost, affirming RRS as an efficient approach to enhance LRM performance without increasing model parameters or requiring retraining.

4.8. Ablation Study of RRS Components

To precisely understand the contribution of each distinct phase within RRS, we conducted an ablation study on the DeepSeek-R1-Distill-Qwen-1.5B model. We compared the full RRS implementation against two ablated variants:

- **RRS (w/o Self-Correction)**: In this variant, the model undergoes the Initial Reasoning Phase and the Self-Diagnosis Phase (generating R_0, A_0, C), but the final answer is taken directly from the initial reasoning A_0. This isolates the effect of merely diagnosing potential errors without explicitly correcting them, primarily serving to quantify the diagnostic capability rather than accuracy improvement based on correction. For Pass@1, this variant will align with BASE.
- **RRS (Direct Revision)**: Here, the model performs the Initial Reasoning Phase (R_0, A_0) and then proceeds directly to the Self-Correction Phase, skipping the explicit '`[CRITIQUE]`' token and the generation of critique C. Instead, the model is prompted directly to revise its initial output based on the original problem and its initial attempt: $M(\text{context} = P \cdot R_0 \cdot A_0 \cdot \texttt{[REVISE]})$. This variant assesses the value of an explicit, separate self-diagnosis step versus an implicit, combined diagnosis-and-correction process.

Table 2 presents the results of this ablation study.

Table 2. Ablation study of RRS components on DeepSeek-R1-Distill-Qwen-1.5B across six benchmarks.

Model & Method	Benchmark	Pass@1 (%)	#Tk (avg. tokens)
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	AIME24	23.3	7280
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	AMC23	57.5	5339
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	Minerva-Math	32.0	4935
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	LiveCode	17.8	6990
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	Olympiad	38.8	5999
RRS (w/o Self-Correction)	AIME24	23.3 (+0.0)	7050
RRS (w/o Self-Correction)	AMC23	57.5 (+0.0)	5400
RRS (w/o Self-Correction)	Minerva-Math	32.0 (+0.0)	4990
RRS (w/o Self-Correction)	LiveCode	17.8 (+0.0)	6850
RRS (w/o Self-Correction)	Olympiad	38.8 (+0.0)	5850
RRS (Direct Revision)	AIME24	27.5 (+4.2)	6900
RRS (Direct Revision)	AMC23	62.0 (+4.5)	5480
RRS (Direct Revision)	Minerva-Math	29.5 (-2.5)	4150
RRS (Direct Revision)	LiveCode	19.5 (+1.7)	6700
RRS (Direct Revision)	Olympiad	40.5 (+1.7)	5700
Full RRS	AIME24	**31.5 (+8.2)**	7150
Full RRS	AMC23	**66.5 (+9.0)**	5600
Full RRS	Minerva-Math	**30.0 (-2.0)**	4250
Full RRS	LiveCode	**21.0 (+3.2)**	6800
Full RRS	Olympiad	**41.8 (+3.0)**	5800

 doi:10.20944/preprints202512.1743.v1

The ablation results underscore the importance of both the Self-Diagnosis and Self-Correction phases. As expected, RRS (w/o Self-Correction) yields the same Pass@1 as BASE because the final answer is still derived from the initial, uncorrected output. However, its slightly higher token count reflects the overhead of generating the critique, even if unused for correction. This variant confirms that merely observing errors internally, without acting upon them, does not improve output accuracy.

More importantly, RRS (Direct Revision) shows an improvement over BASE across most benchmarks (e.g., +4.2% on AIME24, +4.5% on AMC23). This indicates that even without an explicit self-diagnosis step, simply prompting the model to "revise" its initial attempt can lead to better outcomes. The model implicitly performs some form of diagnosis during this revision. However, the Full RRS consistently and significantly outperforms RRS (Direct Revision) across all positive gain benchmarks (e.g., +4.0% additional gain on AIME24 compared to Direct Revision, +4.5% on AMC23). This stark difference highlights the critical role of the explicit Self-Diagnosis phase and the '`[CRITIQUE]`' token. By forcing the LRM to articulate its identified flaws (C), the subsequent Self-Correction phase becomes much more targeted and effective, leading to a superior final answer. The explicit generation of a critique provides a structured internal feedback loop that is more impactful than an implicit revision process. The token counts also support this, with RRS (Direct Revision) being higher than BASE but lower than Full RRS, aligning with the architectural design of each variant.

4.9. Analysis of Error Types Corrected by RRS

To gain deeper qualitative insights into RRS's effectiveness, we analyzed the types of errors that were successfully identified and corrected by the system, particularly focusing on instances where BASE or CoD failed, but RRS provided the correct solution. This analysis was primarily conducted on a subset of mathematical reasoning (AIME24) and code generation (LiveCode) problems. We observed several recurring error categories that RRS was particularly adept at addressing:

- **Logical Inconsistencies and Fallacies:** In complex mathematical and scientific reasoning, LRMs often fall into traps of logical leaps or flawed deductions. RRS's self-diagnosis phase frequently identified these instances, flagging statements like "The logical step from X to Y is not justified" or "This argument relies on an unstated assumption that may not hold." The subsequent self-correction then focused on establishing rigorous connections or rectifying the erroneous logic.
- **Calculation Errors:** Simple arithmetic or algebraic mistakes are common in LRM outputs, especially in multi-step problems. The '`[CRITIQUE]`' phase in RRS was often able to pinpoint specific numerical errors, such as "Error in line 5: 3 * 7 is 21, not 24." This explicit identification allowed the '`[REVISE]`' phase to recalculate and correct these precise points, leading to accurate final answers.
- **Misinterpretation of Problem Constraints:** Problems, particularly in code generation or competitive programming, often come with subtle constraints or edge cases. Initial reasoning might overlook these. RRS critiques often included statements like "Did I consider all edge cases for input N?" or "The problem specifies X, but my solution implicitly assumes Y." The self-correction phase then adjusted the code or reasoning to align with all problem requirements. **Incomplete Reasoning or Omissions:** Sometimes, the initial reasoning (R_0) might be truncated or skip crucial intermediate steps, leading to an unsupported or incorrect answer. The '`[CRITIQUE]`' phase served to identify these gaps, prompting reflections such as "More detailed proof is needed for statement Z" or "I need to explicitly show how this step follows from previous ones." The self-correction then filled in these missing logical connections, resulting in a complete and verifiable solution.
- **Sub-optimal Solutions (Refinement):** Beyond outright errors, RRS also showed capabilities in identifying sub-optimal approaches. In code generation, for instance, a critique might suggest "This algorithm has quadratic complexity; a linear time solution might be possible." The '`[REVISE]`' phase would then attempt to refactor the code for better efficiency or elegance.

The ability of RRS to detect and correct such a diverse range of errors underscores the power of its structured reflective cycle. By systematically breaking down the meta-cognitive process into

 doi:10.20944/preprints202512.1743.v1

distinct diagnosis and correction phases, RRS enables LRMs to move beyond mere pattern matching and engage in a deeper, more robust form of self-scrutiny. This targeted error identification, explicitly generated by the model itself, serves as a highly effective internal feedback mechanism, directly contributing to the enhanced accuracy and robustness observed in our quantitative and human evaluations.

5. Conclusion

This paper introduced the Reflective Reasoning System (RRS), a novel inference-time framework designed to imbue Large Reasoning Models (LRMs) with sophisticated self-diagnosis and self-correction capabilities, thereby significantly enhancing their reasoning accuracy and reliability. Addressing LRMs' inherent limitations in robustness, RRS orchestrates a structured "think-reflect-correct" cycle, mimicking human meta-cognition, through the strategic insertion of '[CRITIQUE]' and '[REVISE]' control tokens. Our comprehensive experimental evaluation yielded compelling evidence for RRS's effectiveness, consistently achieving substantial improvements in Pass@1 accuracy across diverse and challenging benchmarks, including mathematical reasoning, code generation, and scientific problem-solving, and generally outperforming competitive inference-time baselines. Further analysis confirmed that these explicit meta-cognitive tokens effectively activate the LRM's inherent reflective capabilities, transforming a unidirectional reasoning process into a recursive, introspective one. Human evaluation validated higher logical coherence and accuracy, while scalability analysis revealed benefits across various model sizes. An ablation study critically highlighted the necessity of both explicit self-diagnosis and self-correction phases. In conclusion, RRS represents a significant step towards developing more robust and autonomous LRMs, offering a practical and powerful paradigm for enhancing problem-solving accuracy and trustworthiness without architectural modifications or retraining, and paving the way for future meta-cognitive AI systems.

References

1. Mishra, S., Mitra, A., Varshney, N., Sachdeva, B., Clark, P., Baral, C., and Kalyan, A., "NumGLUE: A Suite of Fundamental yet Challenging Mathematical Reasoning Tasks," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2022, pp. 3505–3523. https://doi.org/10.18653/v1/2022.acl-long.246.
2. Zhou, Y., Shen, J., and Cheng, Y., "Weak to strong generalization for large language models with multi-capabilities," *The Thirteenth International Conference on Learning Representations*, 2025.
3. Zhou, Y., Li, X., Wang, Q., and Shen, J., "Visual In-Context Learning for Large Vision-Language Models," *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Association for Computational Linguistics, 2024, pp. 15890–15902.
4. Liu, Y., Yu, R., Yin, F., Zhao, X., Zhao, W., Xia, W., and Yang, Y., "Learning quality-aware dynamic memory for video object segmentation," *European Conference on Computer Vision*, Springer, 2022, pp. 468–486.
5. Liu, Y., Bai, S., Li, G., Wang, Y., and Tang, Y., "Open-vocabulary segmentation with semantic-assisted calibration," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3491–3500.
6. Liu, Y., Zhang, C., Wang, Y., Wang, J., Yang, Y., and Tang, Y., "Universal segmentation at arbitrary granularity with language instruction," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3459–3469.
7. Huang, J., Huang, Y., Liu, J., Zhou, D., Liu, Y., and Chen, S., "Dual-Schedule Inversion: Training-and Tuning-Free Inversion for Real Image Editing," *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2025, pp. 660–669.
8. Huang, Y., Huang, J., Liu, Y., Yan, M., Lv, J., Liu, J., Xiong, W., Zhang, H., Cao, L., and Chen, S., "Diffusion model-based image editing: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
9. Huang, Y., Huang, J., Liu, J., Yan, M., Dong, Y., Lv, J., Chen, C., and Chen, S., "Wavedm: Wavelet-based diffusion models for image restoration," *IEEE Transactions on Multimedia*, Vol. 26, 2024, pp. 7058–7073.
10. Zheng, L., Tian, Z., He, Y., Liu, S., Chen, H., Yuan, F., and Peng, Y., "Enhanced mean field game for interactive decision-making with varied stylish multi-vehicles," *arXiv preprint arXiv:2509.00981*, 2025.

 doi:10.20944/preprints202512.1743.v1

11. Lin, Z., Tian, Z., Lan, J., Zhao, D., and Wei, C., "Uncertainty-Aware Roundabout Navigation: A Switched Decision Framework Integrating Stackelberg Games and Dynamic Potential Fields," *IEEE Transactions on Vehicular Technology*, 2025, pp. 1–13. https://doi.org/10.1109/TVT.2025.3638264.
12. Huang, S., et al., "AI-Driven Early Warning Systems for Supply Chain Risk Detection: A Machine Learning Approach," *Academic Journal of Computing & Information Science*, Vol. 8, No. 9, 2025, pp. 92–107.
13. Huang, S., "Measuring Supply Chain Resilience with Foundation Time-Series Models," *European Journal of Engineering and Technologies*, Vol. 1, No. 2, 2025, pp. 49–56.
14. Ren, L., et al., "Real-time Threat Identification Systems for Financial API Attacks under Federated Learning Framework," *Academic Journal of Business & Management*, Vol. 7, No. 10, 2025, pp. 65–71.
15. Zhu, P., Han, F., and Deng, H., "Flexible multi-generator model with fused spatiotemporal graph for trajectory prediction," *IET Conference Proceedings CP874*, Vol. 2023, IET, 2023, pp. 417–422.
16. Zhu, P., Zhao, S., Han, F., and Deng, H., "BEAVP: A Bidirectional Enhanced Adversarial Model for Video Prediction," *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, IEEE, 2024, pp. 1–8.
17. Zhu, P., "An Empirical Comparative Study of Classical Dimensionality Reduction Methods: MDS, Isomap, and LLE," 2025.
18. Zhou, Y., Geng, X., Shen, T., Tao, C., Long, G., Lou, J.-G., and Shen, J., "Thread of thought unraveling chaotic contexts," *arXiv preprint arXiv:2311.08734*, 2023.
19. Imani, S., Du, L., and Shrivastava, H., "MathPrompter: Mathematical Reasoning using Large Language Models," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, Association for Computational Linguistics, 2023, pp. 37–42. https://doi.org/10.18653/v1/2023.acl-industry.4.
20. Qiao, S., Ou, Y., Zhang, N., Chen, X., Yao, Y., Deng, S., Tan, C., Huang, F., and Chen, H., "Reasoning with Language Model Prompting: A Survey," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023, pp. 5368–5393. https://doi.org/10.18653/v1/2023.acl-long.294.
21. Paranjape, B., Michael, J., Ghazvininejad, M., Hajishirzi, H., and Zettlemoyer, L., "Prompting Contrastive Explanations for Commonsense Reasoning Tasks," *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, 2021, pp. 4179–4192. https://doi.org/10.18653/v1/2021.findings-acl.366.
22. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P., "Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023, pp. 2609–2634. https://doi.org/10.18653/v1/2023.acl-long.147.
23. Li, Z., Jin, X., Guan, S., Li, W., Guo, J., Wang, Y., and Cheng, X., "Search from History and Reason for Future: Two-stage Reasoning on Temporal Knowledge Graphs," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021, pp. 4732–4743. https://doi.org/10.18653/v1/2021.acl-long.365.
24. Cao, S., Shi, J., Pan, L., Nie, L., Xiang, Y., Hou, L., Li, J., He, B., and Zhang, H., "KQA Pro: A Dataset with Explicit Compositional Programs for Complex Question Answering over Knowledge Base," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2022, pp. 6101–6119. https://doi.org/10.18653/v1/2022.acl-long.422.
25. Zhong, W., Cui, R., Guo, Y., Liang, Y., Lu, S., Wang, Y., Saied, A., Chen, W., and Duan, N., "AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models," *Findings of the Association for Computational Linguistics: NAACL 2024*, Association for Computational Linguistics, 2024, pp. 2299–2314. https://doi.org/10.18653/v1/2024.findings-naacl.149.
26. Mishra, S., Finlayson, M., Lu, P., Tang, L., Welleck, S., Baral, C., Rajpurohit, T., Tafjord, O., Sabharwal, A., Clark, P., and Kalyan, A., "LILA: A Unified Benchmark for Mathematical Reasoning," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2022, pp. 5807–5832. https://doi.org/10.18653/v1/2022.emnlp-main.392.
27. Tian, Z., Lin, Z., Zhao, D., Zhao, W., Flynn, D., Ansari, S., and Wei, C., "Evaluating scenario-based decision-making for interactive autonomous driving using rational criteria: A survey," *arXiv preprint arXiv:2501.01886*, 2025.

 doi:10.20944/preprints202512.1743.v1

28. Zhang, F., Chen, H., Zhu, Z., Zhang, Z., Lin, Z., Qiao, Z., Zheng, Y., and Wu, X., "A survey on foundation language models for single-cell biology," *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 528–549.
29. Zhang, F., Liu, T., Zhu, Z., Wu, H., Wang, H., Zhou, D., Zheng, Y., Wang, K., Wu, X., and Heng, P.-A., "CellVerse: Do Large Language Models Really Understand Cell Biology?" *arXiv preprint arXiv:2505.07865*, 2025.
30. Zhang, F., Liu, T., Chen, Z., Peng, X., Chen, C., Hua, X.-S., Luo, X., and Zhao, H., "Semi-supervised knowledge transfer across multi-omic single-cell data," *Advances in Neural Information Processing Systems*, Vol. 37, 2024, pp. 40861–40891.
31. Sun, H., Zhong, J., Ma, Y., Han, Z., and He, K., "TimeTraveler: Reinforcement Learning for Temporal Knowledge Graph Forecasting," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021, pp. 8306–8319. https://doi.org/10.18653/v1/2021.emnlp-main.655.
32. Zuo, X., Cao, P., Chen, Y., Liu, K., Zhao, J., Peng, W., and Chen, Y., "Improving Event Causality Identification via Self-Supervised Representation Learning on External Causal Statement," *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, 2021, pp. 2162–2172. https://doi.org/10.18653/v1/2021.findings-acl.190.
33. Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H., "Self-Instruct: Aligning Language Models with Self-Generated Instructions," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023, pp. 13484–13508. https://doi.org/10.18653/v1/2023.acl-long.754.
34. Hu, X., Zhang, C., Yang, Y., Li, X., Lin, L., Wen, L., and Yu, P. S., "Gradient Imitation Reinforcement Learning for Low Resource Relation Extraction," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021, pp. 2737–2746. https://doi.org/10.18653/v1/2021.emnlp-main.216.
35. Wang, B., Deng, X., and Sun, H., "Iteratively Prompt Pre-trained Language Models for Chain of Thought," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2022, pp. 2714–2730. https://doi.org/10.18653/v1/2022.emnlp-main.174.
36. Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., and Wei, F., "Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers," *Findings of the Association for Computational Linguistics: ACL 2023*, Association for Computational Linguistics, 2023, pp. 4005–4019. https://doi.org/10.18653/v1/2023.findings-acl.247.
37. Taranovsky, D., "Reflective Cardinals," *arXiv preprint arXiv:1203.2270v5*, 2012.
38. Zhang, Y., Li, Z., Bao, Z., Li, J., Zhang, B., Li, C., Huang, F., and Zhang, M., "MuCGEC: a Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction," *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2022, pp. 3118–3130. https://doi.org/10.18653/v1/2022.naacl-main.227.
39. Yang, W., Lin, Y., Li, P., Zhou, J., and Sun, X., "RAP: Robustness-Aware Perturbations for Defending against Backdoor Attacks on NLP Models," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021, pp. 8365–8381. https://doi.org/10.18653/v1/2021.emnlp-main.659.
40. Zhang, J., Dong, R., Wang, H., Ning, X., Geng, H., Li, P., He, X., Bai, Y., Malik, J., Gupta, S., and Zhang, H., "AlphaOne: Reasoning Models Thinking Slow and Fast at Test Time," *CoRR*, 2025. https://doi.org/10.48550/ARXIV.2505.24863.