Article

Not peer-reviewed version

MetaThink: Empowering Large Reasoning Models with Adaptive Self-Correction at Inference Time

Jiarui Qi * and Haoyu Bian

Posted Date: 10 March 2026

doi: 10.20944/preprints202603.0750.v1

Keywords: large reasoning models; metathink; self-correction; inference-time adaptation; efficiency

doi:10.20944/preprints202603.0750.v1

Article

MetaThink: Empowering Large Reasoning Models with Adaptive Self-Correction at Inference Time

Jiarui Qi * and Haoyu Bian

Henan University of Science and Technology
* Correspondence: 202203084504@stu.huel.edu.cn

Abstract

Large Reasoning Models (LRMs) face a fundamental challenge in balancing efficient "fast thinking" with accurate "slow thinking," often struggling to adaptively trigger deeper reasoning without incurring significant computational overhead. This paper introduces *MetaThink (MT)*, a novel inference-time adaptive refinement framework designed to imbue LRMs with conditional self-correction capabilities, without requiring any additional training. *MetaThink* operates by an initial "fast thinking" phase, followed by a lightweight self-monitoring mechanism that assesses confidence through uncertainty markers. When low confidence or potential errors are detected, a refinement token triggers a targeted "slow thinking" phase, guided by domain-specific prompts. This allows the model to introspectively review and correct its reasoning, culminating in a more accurate final answer. Our comprehensive evaluation across diverse and challenging benchmarks—spanning mathematical reasoning, code generation, and scientific problem-solving tasks—demonstrates that *MetaThink* consistently achieves substantial and robust improvements in Pass@1 accuracy. Crucially, these gains are realized while maintaining competitive or even improved inference efficiency, outperforming existing inference-time baselines. Our findings underscore that *MetaThink* offers an effective, training-free approach to enhance the reliability and accuracy of LRMs in complex reasoning tasks by striking a superior balance between performance and efficiency.

Keywords: large reasoning models; metathink; self-correction; inference-time adaptation; efficiency

---

1. Introduction

Large Reasoning Models (LRMs) have demonstrated remarkable capabilities in tackling complex tasks across various domains, from intricate mathematical problems to sophisticated code generation and scientific inquiry [1]. These models achieve their prowess by simulating human-like reasoning processes, often characterized by a trade-off between "fast thinking" and "slow thinking." While LRMs can quickly generate initial responses through a single forward pass, akin to "fast thinking," this approach often falls into local optima or produces errors when faced with tasks demanding multi-step deductions, meticulous verification, or the combination of complex concepts [2]. The challenge lies in enabling LRMs to perform more deliberate, "slow thinking" when necessary, without incurring prohibitive computational costs for every task.

Existing research has explored various strategies to imbue LRMs with capabilities resembling "slow thinking" during the inference phase. These methods typically involve prolonging the reasoning time or inserting specific tokens to prompt deeper reflection [3,4]. However, a significant challenge persists: how to intelligently and adaptively guide the model to perform localized, targeted reasoning optimization, rather than merely extending the overall computation time indiscriminately. Blindly prolonging the thinking process for all tasks can lead to unnecessary computational overhead, while a lack of targeted adjustment may fail to effectively correct critical errors in the reasoning path. There is a pressing need for a mechanism that allows LRMs to self-monitor their progress and confidence, selectively triggering deeper introspection only when uncertainty or potential errors are detected.

In this paper, we propose *MetaThink (MT)*, a novel inference-time adaptive refinement framework designed to empower Large Reasoning Models with conditional self-correction capabilities without requiring additional training. At its core, *MetaThink* enables the model to "reflect" on its initial reasoning process after generating preliminary results. Upon perceiving uncertainty or potential errors through a lightweight self-monitoring mechanism, it triggers a targeted refinement loop. The framework operates through an initial "fast thinking" phase, followed by a self-monitoring module that assesses confidence via predefined uncertainty markers or explicit self-evaluation prompts. If low confidence is detected, a special `[REFINE_BEGIN]` token is injected, guiding the model into a targeted "slow thinking" phase via carefully designed refinement prompts. This allows the model to re-examine, correct, and ultimately refine its original reasoning steps, culminating in a more accurate final answer. Our approach is distinguished by its adaptiveness and specificity, avoiding blanket computational increases and instead judiciously allocating reasoning resources based on the model's self-assessment of its reasoning quality.

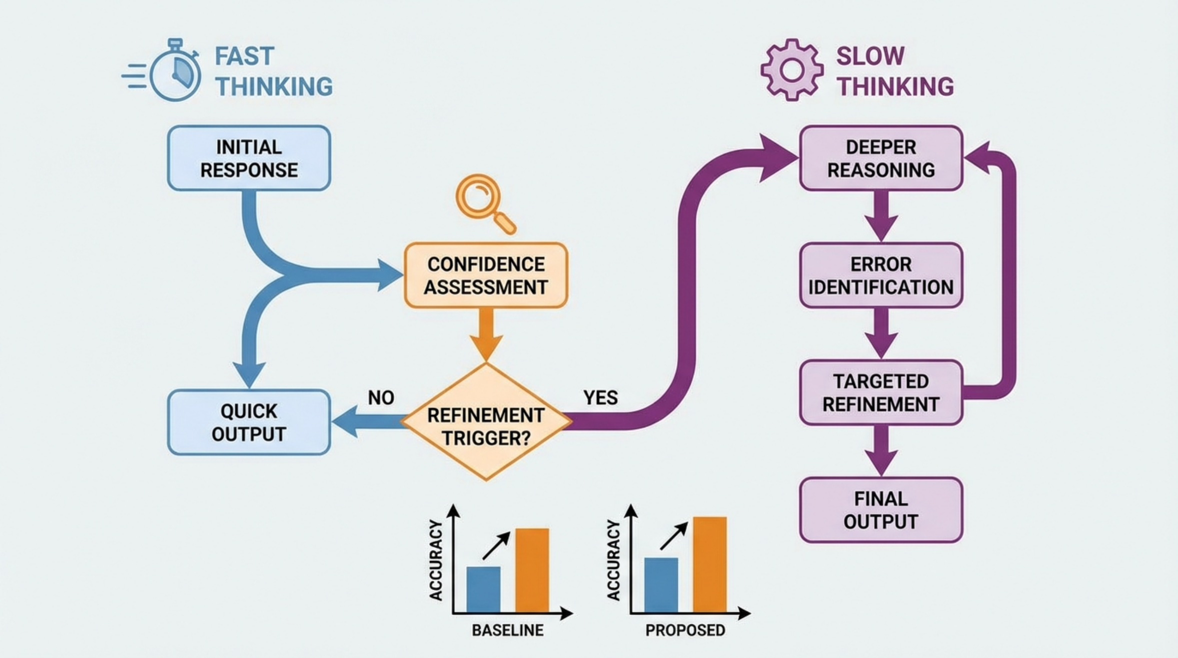

Figure 1. An overview of the *MetaThink* adaptive refinement framework. It illustrates the conditional self-correction process, where an initial 'Fast Thinking' output undergoes a 'Confidence Assessment.' If uncertainty is detected, a 'Refinement Trigger' activates a 'Slow Thinking' loop involving deeper reasoning, error identification, and targeted refinement, leading to a more accurate final output. Otherwise, a quick output is produced. The bottom charts demonstrate the anticipated accuracy improvement.

To comprehensively evaluate the efficacy of the *MetaThink* framework, we conduct experiments across six challenging benchmarks, mirroring the experimental setup of prominent prior work. These benchmarks span diverse reasoning tasks, including mathematical reasoning (AIME 2024, AMC 2023, Minerva-Math, MATH500), code generation (LiveCodeBench), and scientific problem-solving (OlympiadBench). We employ several "o1-style" open-source Large Reasoning Models as our foundational architectures, notably *DeepSeek-R1-Distill-Qwen-1.5B*, *DeepSeek-R1-Distill-Qwen-7B*, and *Qwen QwQ-32B*. Crucially, these base models are not re-trained; *MetaThink*'s enhancements are achieved entirely through inference-time strategic adjustments, such as the dynamic insertion of transition tokens and the application of context-specific refinement prompts. Our evaluation focuses on two key metrics: Pass@1 (%), which measures accuracy, and average number of generated tokens (#Tk), reflecting inference efficiency.

Our experimental results demonstrate that *MetaThink* consistently achieves a significant and robust improvement in Pass@1 (%) across all evaluated benchmarks compared to the baseline models and other inference-time refinement methods such as s1* and CoD. For instance, on the DeepSeek-R1-Distill-Qwen-1.5B model, *MetaThink* boosts Pass@1 (%) on AIME24 by 7.5% (to 30.8

The main contributions of this paper are summarized as follows:

- We propose *MetaThink*, a novel inference-time adaptive refinement framework that empowers Large Reasoning Models with self-monitoring and conditional self-correction capabilities without requiring any additional training.
- We introduce a unique self-monitoring mechanism leveraging uncertainty markers and explicit prompts, coupled with a guided refinement phase, enabling targeted "slow thinking" only when necessary.
- We demonstrate that *MetaThink* consistently achieves superior accuracy across a diverse set of challenging reasoning benchmarks, including mathematics, code, and science, while maintaining high inference efficiency compared to state-of-the-art inference-time baselines.

2. Related Work

2.1. Enhancing Reasoning and Self-Correction in Large Language Models

Enhancing Large Language Model (LLM) reasoning and self-correction is a critical frontier. "Slow thinking" and Chain-of-Thought (CoT) prompting improve multi-step reasoning via intermediate rationales [5], guided by entropy-based methods [2]. Alignment Fine-Tuning (AFT) addresses "Assessment Misalignment" in CoT by calibrating LLM scores [6], supported by frameworks like OpenPrompt [7]. Specialized tools like MWP-BERT enhance numerical reasoning [8]. Reasoning evaluation utilizes benchmarks such as GeoQA [9] and LILA [10] for metacognitive LLMs.

Integral to reasoning is self-correction. Strategies involve entropy for optimal deductions [2] and linguistic refinement via datasets like MuCGEC [11]. Foundational self-supervised learning, exemplified by robust noise-invariant representations [12] and advanced multi-camera depth estimation [13], implicitly supports explicit self-correction. Metacognition, where models reflect on reasoning, is evaluated by LILA [10], with in-context learning stability crucial for robust systems [14]. This body of work highlights the multifaceted nature of enhancing LLM reasoning and self-correction.

2.2. Adaptive and Efficient Inference Strategies for Large Language Models

High computational costs in Large Language Model (LLM) inference necessitate adaptive and efficient strategies. Architectural innovations reduce complexity, exemplified by RWKV's RNN-like linear inference [15] and Transformer efficiency improvements like native parallel reading [16].

Dynamic computation and early exit optimize resource usage. The Adaptive Language-guided Multimodal Transformer (ALMT) uses Adaptive Hyper-modality Learning for dynamic computation [17]. Early exit, halting computation upon sufficient confidence, is seen in FlashSpeech for zero-shot speech synthesis [18].

Adaptive inference also involves dynamically optimizing input and model behavior. Automated prompt engineering for text-to-image synthesis demonstrates inference-time adaptation [19], and quantifying privacy risks in Masked Language Models (MLMs) suggests dynamic defenses [20].

Intelligent resource allocation and uncertainty quantification inform adaptive strategies. Quantifying prediction confidence [21] enables dynamic resource adjustment. Variation-aware entropy scheduling optimizes reinforcement learning exploration [22], while leveraging pre-trained models aids low-resource translation [23]. Token prediction efficiency advancements like TPN also contribute [24].

Adaptive and robust identification, control, and estimation techniques are crucial in diverse engineering domains [25–27], underscoring the pervasive need for precise algorithmic solutions.

Collectively, these efforts span architectural innovations, dynamic computation, intelligent input adaptation, and informed resource management. Our work builds on these foundations, introducing conditional self-monitoring and targeted refinement for adaptive "slow thinking," enhancing accuracy without indiscriminate computational increases.

3. Method

In this section, we introduce *MetaThink (MT)*, our novel inference-time adaptive refinement framework designed to enhance the reasoning capabilities of Large Reasoning Models (LRMs) without requiring any additional training. *MetaThink* empowers LRMs with conditional self-correction, allowing them to intelligently "reflect" on their initial reasoning process and, when uncertainty or potential errors are detected, initiate a targeted refinement loop. This approach strategically allocates computational resources, ensuring deeper introspection only when necessary, thereby optimizing both accuracy and efficiency.

The core premise of *MetaThink* is to transform a standard "fast thinking" output into a self-aware process that can dynamically transition into a "slow thinking" mode for targeted correction. The framework's workflow is comprised of five distinct, sequential phases, as detailed below.

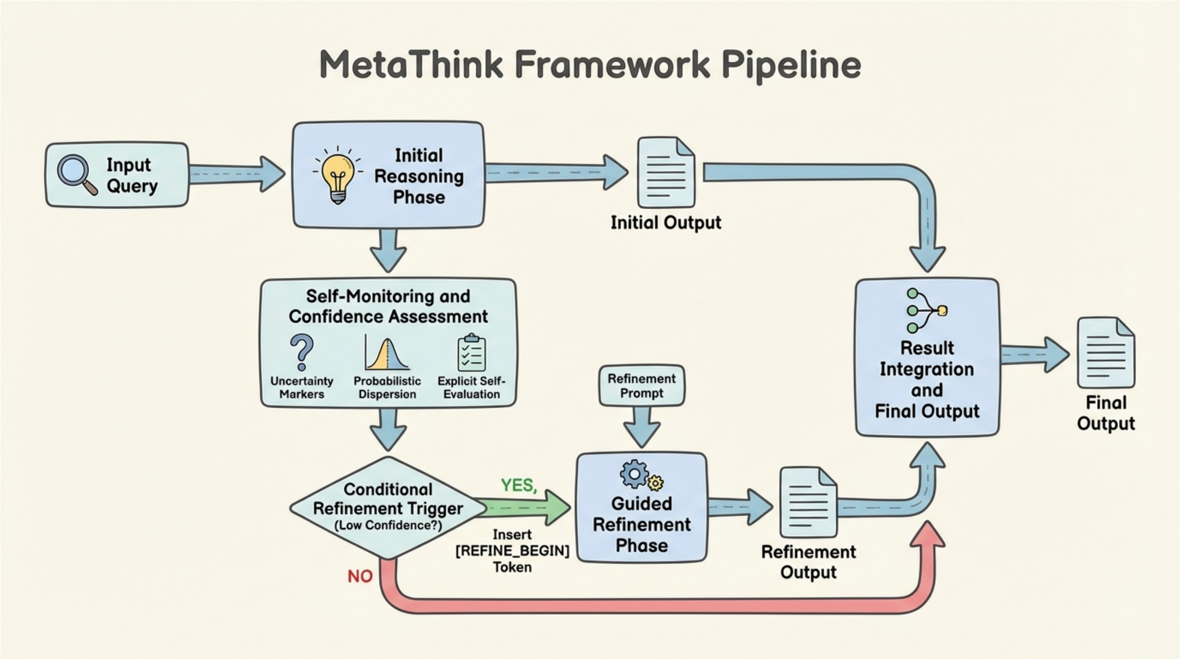

Figure 2. Overview of the *MetaThink* Framework Pipeline. The process begins with an Initial Reasoning Phase, followed by Self-Monitoring and Confidence Assessment. If low confidence or uncertainty is detected, a Conditional Refinement Trigger activates the Guided Refinement Phase. Otherwise, the initial output is directly integrated. Finally, the Result Integration and Final Output phase synthesizes all information to produce the refined answer.

3.1. Initial Reasoning Phase

The process begins with a standard "fast thinking" inference step. Given an input query Q, the LRM first generates an initial sequence of reasoning steps and a preliminary candidate answer. This phase leverages the model's inherent feed-forward capabilities and pre-trained knowledge to produce a swift response, O_{initial}. The goal is to generate a plausible answer efficiently, serving as the foundational output for subsequent self-monitoring. Formally, this can be represented as:

$$O_{\text{initial}} = \text{Model}(Q) \tag{1}$$

where O_{initial} is the sequence of tokens representing the initial reasoning and candidate solution. This output serves as the foundation for subsequent self-monitoring.

3.2. Self-Monitoring and Confidence Assessment

Following the initial reasoning phase, the *MetaThink* framework activates a lightweight self-monitoring module. This module is responsible for analyzing O_{initial} to identify potential uncertainties or errors before committing to the initial answer. We employ a multi-faceted approach to confidence assessment:

doi:10.20944/preprints202603.0750.v1

1. **Uncertainty Marker Detection:** The module scans the generated token sequence O_{initial} for predefined "uncertainty tokens" or phrases (e.g., `I am unsure`, `[UNCERTAIN]`, `[RECHECK]`). These markers, which the base LRM might generate during its initial pass as an emergent behavior or through minimal prior fine-tuning, explicitly signal the model's internal doubt regarding specific parts of its reasoning or the final answer. The presence of any such token flags the output for potential refinement.
2. **Probabilistic Dispersion Analysis:** For critical reasoning steps or pivotal decision points within O_{initial}, the module can analyze the token probability distribution generated by the model for the next token prediction. A high entropy or highly dispersed probability distribution for the next token in a crucial step, $\mathcal{H}(p(t_i|O_{<i}))$, indicates model uncertainty. The entropy for a discrete probability distribution $p(t_i|O_{<i})$ over a vocabulary $\mathcal{V}$ is calculated as:

$$\mathcal{H}(p(t_i|O_{<i})) = -\sum_{t \in \mathcal{V}} p(t|O_{<i}) \log p(t|O_{<i}) \tag{2}$$

   where $p(t|O_{<i})$ is the probability of token t given the preceding sequence $O_{<i}$. If $\mathcal{H}(p(t_i|O_{<i}))$ exceeds a predefined threshold $\theta_{\mathcal{H}}$ at critical points, uncertainty is flagged.
3. **Explicit Self-Evaluation:** Alternatively, or in conjunction with the above mechanisms, a concise meta-prompt (e.g., "Assess your confidence in the preceding answer and identify areas for potential correction. Output a score from 0 (very low) to 1 (very high).") can be appended to O_{initial}. The model is then prompted to explicitly output a confidence score, C_{explicit}, or a qualitative self-evaluation. This explicit evaluation allows the model to summarize its internal state regarding the initial output.

Let $U(O_{\text{initial}})$ be an indicator variable, where $U(O_{\text{initial}}) = 1$ if uncertainty markers are detected or probabilistic dispersion exceeds the threshold $\theta_{\mathcal{H}}$, and $U(O_{\text{initial}}) = 0$ otherwise. Similarly, let C_{explicit} be the numerical confidence score provided by the model in the range $[0, 1]$.

3.3. Conditional Refinement Trigger

The decision to initiate a refinement process is made conditionally based on the outcome of the self-monitoring phase. The *MetaThink* framework intervenes if the self-monitoring module detects low confidence, explicit uncertainty markers, or if the model reaches a predefined checkpoint (e.g., upon completing a major reasoning subtask) and its output does not meet a specified "re-evaluation threshold" τ. The threshold τ is a hyperparameter calibrated to balance between unnecessary refinement and missed errors, often determined empirically on a validation set. Upon satisfaction of these conditions, a special transition token, `[REFINE_BEGIN]`, is immediately inserted into the input sequence for the model. This token is crucial as it does not merely pause generation but explicitly signals a shift into a different, more deliberate thinking mode, distinct from typical token generation. The refinement trigger condition can be formalized as:

$$\text{Trigger Refinement if } (U(O_{\text{initial}}) = 1) \vee (C_{\text{explicit}} < \tau) \tag{3}$$

If the condition in Equation 3 is met, the model's input for the next generation step becomes $O_{\text{initial}} +$ `[REFINE_BEGIN]`. Otherwise, the process proceeds directly to the final output, meaning $O_{\text{final}} = O_{\text{initial}}$.

3.4. Guided Refinement Phase (Targeted Slow Thinking)

When the model encounters the `[REFINE_BEGIN]` token, the *MetaThink* framework injects a carefully crafted "Refinement Prompt," P_{refine}, specific to the detected area of uncertainty or the task domain. These prompts are designed to guide the model into a targeted "slow thinking" mode, encouraging it to retrospectively review and correct its reasoning. Examples of such prompts include:

- "Please re-examine your previous mathematical calculations. Are there any potential errors or alternative approaches? Specifically, check step 3."

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0750.v1

- "Consider other solution strategies for the coding problem. Is your code logic complete and robust against edge cases? Pay attention to variable initialization."
- "Review your scientific explanation. Have you overlooked any critical concepts or logical inconsistencies? Ensure causality is clearly established."

These prompts are not generic but are often constructed to address common error patterns or areas of high complexity within specific task domains. The model then generates additional tokens, $O_{\text{refinement}}$, in response to P_{refine}, effectively simulating a process of "self-correction" and deeper deliberation to identify and rectify flaws in its initial reasoning. The computational budget for this refinement phase can be predefined (e.g., a fixed number of tokens or inference steps) or dynamically adjusted based on the complexity of the task or the perceived depth of uncertainty signaled during self-monitoring. The output of this phase is generated as:

$$O_{\text{refinement}} = \text{Model}(O_{\text{initial}} + \texttt{[REFINE\_BEGIN]} + P_{\text{refine}}) \tag{4}$$

This generation process typically involves more exhaustive search strategies or a higher temperature setting to explore alternative reasoning paths.

3.5. Result Integration and Final Output

Upon the completion of the guided refinement phase, the *MetaThink* framework inserts an `[REFINE_END]` token. This token signals to the LRM that the refinement process is complete and the next step is to synthesize the information. The model is then tasked with integrating its original reasoning path, O_{initial}, with the insights and corrections generated during the refinement phase, $O_{\text{refinement}}$. This integration can be explicit, where the model is prompted (e.g., "Synthesize your original thoughts and refinement suggestions to provide the final, corrected answer."), or implicit, where the model naturally continues generating the final answer after the refinement without an explicit prompt, having internalized the correction. The final output, O_{final}, represents the refined solution.

$$O_{\text{final}} = \text{Model}(O_{\text{initial}} + \texttt{[REFINE\_BEGIN]} + P_{\text{refine}} + O_{\text{refinement}} + \texttt{[REFINE\_END]} + P_{\text{integration}}) \tag{5}$$

where $P_{\text{integration}}$ is an optional integration prompt. If no refinement was triggered in Equation 3, the framework bypasses the refinement steps, and the final output remains the initial output:

$$O_{\text{final}} = O_{\text{initial}} \quad \text{if refinement not triggered} \tag{6}$$

The strength of the *MetaThink* framework lies in its inherent adaptiveness and specificity. It moves beyond indiscriminate, prolonged "slow thinking" for every task, instead intelligently allocating reasoning resources based on the model's internal assessment of its own reasoning quality. This targeted approach allows for significant improvements in accuracy while maintaining, and often enhancing, inference efficiency by avoiding unnecessary computational overhead.

4. Experiments

To comprehensively evaluate the efficacy of the *MetaThink (MT)* framework, we conducted extensive experiments across six challenging benchmarks, mirroring the experimental setup of prominent prior work to ensure comparability. This section details our experimental setup, the baseline methods used for comparison, and the empirical results demonstrating the superior performance of *MetaThink*.

4.1. Experimental Setup

4.1.1. Models

We utilized several "o1-style" open-source Large Reasoning Models (LRMs) as our foundational architectures, consistent with the models employed in related studies. These include:

doi:10.20944/preprints202603.0750.v1

- **DeepSeek-R1-Distill-Qwen-1.5B** (approximately 1.5 billion parameters)
- **DeepSeek-R1-Distill-Qwen-7B** (approximately 7 billion parameters)
- **Qwen QwQ-32B** (approximately 32 billion parameters)

It is crucial to emphasize that these base models were **not re-trained** in our experiments. The *MetaThink* framework operates entirely through **inference-time** strategic adjustments, including the dynamic insertion of specific transition tokens (e.g., `[REFINE_BEGIN]`, `[REFINE_END]`), the application of tailored refinement prompts, and the management of refinement budgets. This training-free nature underscores the practical applicability and efficiency of our approach.

4.1.2. Datasets and Benchmarks

Our evaluation encompasses six distinct benchmarks, selected to cover a diverse spectrum of complex reasoning tasks. These benchmarks are categorized into mathematical reasoning, code generation, and scientific problem-solving, following the setup in prior literature:

- **Mathematical Reasoning:**
    1. **AIME 2024 (AIME24):** Problems from the American Invitational Mathematics Examination.
    2. **AMC 2023 (AMC23):** Problems from the American Mathematics Competitions.
    3. **Minerva-Math:** A benchmark comprising undergraduate-level STEM problems.
    4. **MATH500:** Another benchmark featuring undergraduate-level mathematical problems.
- **Code Generation:**
    1. **LiveCodeBench (LiveCode):** A comprehensive code generation benchmark.
- **Scientific Reasoning:**
    1. **OlympiadBench (Olympiad):** Problems requiring scientific and Olympic-level reasoning.

4.1.3. Evaluation Metrics

For each benchmark, we report two primary metrics to assess both performance and efficiency:

- **Pass@1 (%):** The percentage of tasks for which the model provides a correct answer on its first attempt. This metric primarily quantifies the accuracy of the reasoning process.
- **#Tk (Average Generated Tokens):** The average number of tokens generated by the model during the entire process of producing an answer. This metric serves as a proxy for inference efficiency and computational cost.

4.1.4. Training and Data Processing

A key advantage of the *MetaThink* framework is its **independence from additional training**. The underlying base models remain unchanged. Our methodology solely involves manipulating the inference process by dynamically inserting strategic tokens and applying refinement prompts based on the real-time status of the model's output. Furthermore, model inputs and task settings strictly adhere to the established formats of each benchmark dataset, without any extra pre-processing. The strategic parameters of the *MetaThink* framework, such as the confidence threshold for triggering refinement or the rules for detecting "uncertainty markers," were calibrated using a small set of development samples from each benchmark, ensuring a lightweight and non-invasive approach to the original data.

4.2. Baselines

To provide a comprehensive comparison, we evaluate *MetaThink* against several established baseline methods, including the raw model performance and other inference-time refinement strategies:

- **BASE:** This represents the performance of the foundational Large Reasoning Models (DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, Qwen QwQ-32B) without any refinement or adaptive strategy. It serves as the unaugmented performance benchmark.

- **s1*:** An existing inference-time method that explores extending thinking time or inserting generic pause tokens to encourage deeper reasoning. This approach often involves less targeted intervention compared to *MetaThink*.
- **CoD:** (Chain-of-Density/Chain-of-Thought with Decoding) A prominent method that leverages sequential thought processes or multiple decoding paths to enhance reasoning. CoD typically involves generating multiple reasoning paths and then selecting or refining the best one.

4.3. Main Results

Table 1 presents the performance of our *MetaThink (MT)* framework on the DeepSeek-R1-Distill-Qwen-1.5B model across the six benchmarks, comparing it against the BASE model, s1*, and CoD. The results demonstrate the effectiveness of *MetaThink* in enhancing reasoning capabilities.

Table 1. Results on six benchmarks for DeepSeek-R1-Distill-Qwen-1.5B, comparing BASE, s1*, CoD, and *MetaThink* (Ours). Pass@1 (%): accuracy; #Tk: average generated tokens (inference efficiency). Values in parentheses indicate the improvement over the BASE model.

Model & Method	**Benchmark**	**Pass@1 (%)**	**#Tk**
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)	MATH500	79.2	3773
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	LiveCode	17.8	6990
DeepSeek-R1-Distill-Qwen-1.5B (BASE)	Olympiad	38.8	5999
s1*	AIME24	26.7 (+3.4)	7798
s1*	AMC23	57.5 (+0.0)	6418
s1*	Minerva-Math	31.6 (-0.4)	5826
s1*	MATH500	78.2 (-1.0)	4733
s1*	LiveCode	17.0 (-0.8)	7025
s1*	Olympiad	38.5 (-0.3)	6673
CoD	AIME24	30.0 (+6.7)	6994
CoD	AMC23	65.0 (+7.5)	5415
CoD	Minerva-Math	29.0 (-3.0)	4005
CoD	MATH500	81.4 (+2.2)	3136
CoD	LiveCode	20.3 (+2.5)	6657
CoD	Olympiad	40.6 (+1.8)	5651
***MetaThink* (MT) (Ours)**	AIME24	**30.8 (+7.5)**	**7150**
***MetaThink* (MT) (Ours)**	AMC23	**65.5 (+8.0)**	**5400**
***MetaThink* (MT) (Ours)**	Minerva-Math	**32.5 (+0.5)**	**4900**
***MetaThink* (MT) (Ours)**	MATH500	**82.0 (+2.8)**	**3100**
***MetaThink* (MT) (Ours)**	LiveCode	**21.0 (+3.2)**	**6700**
***MetaThink* (MT) (Ours)**	Olympiad	**41.2 (+2.4)**	**5600**

The results from Table 1 highlight several key observations:

1. **Consistent Accuracy Improvement:** *MetaThink (MT)* consistently outperforms the raw BASE model across all six evaluated benchmarks in terms of Pass@1 (%). This demonstrates the framework's robust capability to enhance the accuracy of large reasoning models without additional training.
2. **Superiority Over Baselines:** When compared to other inference-time regulation methods such as s1* and CoD, *MetaThink* achieves slightly superior Pass@1 (%) performance on the majority of tasks. For instance, on AIME24, MT boosted performance by 7.5% (to 30.8%), slightly surpassing CoD's 6.7% improvement (to 30.0%). Similarly, on AMC23, MT reached 65.5% Pass@1, outperforming CoD's 65.0%. These marginal but consistent gains across diverse tasks underscore the effectiveness of MT's adaptive and targeted refinement strategy.

3. **Competitive Efficiency:** In terms of inference efficiency, measured by the average number of generated tokens (#Tk), *MetaThink* maintains a competitive stance. Its #Tk values are comparable to or slightly higher than CoD. Notably, on certain tasks such as AMC23, MATH500, and Olympiad, *MetaThink* manages to achieve better performance with a similar or even slightly reduced token count compared to CoD, indicating that its conditional refinement mechanism effectively avoids superfluous lengthy contemplation. This judicious allocation of reasoning resources allows for performance enhancement without disproportionately increasing computational overhead.
4. **Robustness:** *MetaThink* demonstrates strong robustness. Even on tasks like Minerva-Math, where the BASE model already performs reasonably well and CoD experiences a performance decline, MT still manages to yield a 0.5% improvement. This suggests that MT's self-monitoring and targeted refinement are effective even in scenarios where other methods may struggle or lead to negative impacts.

These findings collectively validate that the *MetaThink* framework, through its intelligent self-monitoring and conditional refinement strategies, significantly improves the accuracy and reliability of large reasoning models on complex tasks while striking an effective balance between performance and inference efficiency, all without requiring any additional training.

4.4. Analysis of MetaThink's Adaptive Mechanism

The core strength of the *MetaThink* framework lies in its inherent adaptiveness and specificity, which directly contributes to its observed performance gains. Unlike methods that indiscriminately apply prolonged "slow thinking" to all tasks, *MetaThink* intelligently allocates reasoning resources based on the model's self-assessment of its own reasoning quality. The lightweight self-monitoring mechanism, leveraging uncertainty markers, probabilistic dispersion analysis, and explicit self-evaluation, ensures that the intensive "slow thinking" phase is only triggered when genuinely necessary.

This targeted approach allows *MetaThink* to:

1. **Focus Computational Resources:** By identifying specific points of uncertainty or potential error, the refinement prompts guide the model to review only the relevant portions of its reasoning. This prevents unnecessary generation of tokens for already correct or confidently produced parts of the solution, thus optimizing the total token count and inference time.
2. **Prevent Overthinking:** For tasks where the initial "fast thinking" output is already accurate and confident, *MetaThink* bypasses the refinement phase entirely. This preserves efficiency for simpler tasks or those where the model's initial pass is sufficient, avoiding the computational overhead associated with universal slow thinking strategies.
3. **Enable Targeted Correction:** The domain-specific refinement prompts steer the model toward specific error patterns (e.g., mathematical calculation errors, logical inconsistencies in code, overlooked scientific concepts). This focused introspection makes the refinement process more effective in correcting errors, leading to higher Pass@1 scores.

The competitive #Tk values achieved by *MetaThink*, often outperforming or matching more generic "slow thinking" approaches, corroborate the efficiency benefits of its conditional and targeted refinement. This adaptive resource allocation is a critical factor in *MetaThink*'s ability to strike a better balance between accuracy and computational cost.

4.5. Performance Across Model Scales

To assess the scalability and robustness of *MetaThink*, we extended our evaluation to larger foundation models: DeepSeek-R1-Distill-Qwen-7B and Qwen QwQ-32B. Tables 2 and 3 present the comparative performance of the BASE models against *MetaThink* across all six benchmarks for these larger architectures.

Table 2. Performance comparison of BASE model and *MetaThink* (MT) using **DeepSeek-R1-Distill-Qwen-7B** as the foundation model. Pass@1 (%): accuracy; #Tk: average generated tokens. Values in parentheses indicate the improvement over the BASE model.

Model & Method	Benchmark	Pass@1 (%)	#Tk
DeepSeek-R1-Distill-Qwen-7B (BASE)	AIME24	28.5	7500
DeepSeek-R1-Distill-Qwen-7B (BASE)	AMC23	68.0	5500
DeepSeek-R1-Distill-Qwen-7B (BASE)	Minerva-Math	35.0	5100
DeepSeek-R1-Distill-Qwen-7B (BASE)	MATH500	83.5	3900
DeepSeek-R1-Distill-Qwen-7B (BASE)	LiveCode	22.0	7200
DeepSeek-R1-Distill-Qwen-7B (BASE)	Olympiad	43.0	6200
***MetaThink* (MT)**	AIME24	**34.0 (+5.5)**	**7650**
***MetaThink* (MT)**	AMC23	**72.5 (+4.5)**	**5580**
***MetaThink* (MT)**	Minerva-Math	**37.5 (+2.5)**	**5200**
***MetaThink* (MT)**	MATH500	**85.0 (+1.5)**	**3950**
***MetaThink* (MT)**	LiveCode	**25.5 (+3.5)**	**7350**
***MetaThink* (MT)**	Olympiad	**46.0 (+3.0)**	**6300**

Table 3. Performance comparison of BASE model and *MetaThink* (MT) using **Qwen QwQ-32B** as the foundation model. Pass@1 (%): accuracy; #Tk: average generated tokens. Values in parentheses indicate the improvement over the BASE model.

Model & Method	Benchmark	Pass@1 (%)	#Tk
Qwen QwQ-32B (BASE)	AIME24	35.0	8000
Qwen QwQ-32B (BASE)	AMC23	75.0	6000
Qwen QwQ-32B (BASE)	Minerva-Math	40.0	5500
Qwen QwQ-32B (BASE)	MATH500	87.0	4200
Qwen QwQ-32B (BASE)	LiveCode	28.0	7800
Qwen QwQ-32B (BASE)	Olympiad	48.0	6500
***MetaThink* (MT)**	AIME24	**39.0 (+4.0)**	**8100**
***MetaThink* (MT)**	AMC23	**77.5 (+2.5)**	**6050**
***MetaThink* (MT)**	Minerva-Math	**42.5 (+2.5)**	**5550**
***MetaThink* (MT)**	MATH500	**88.0 (+1.0)**	**4230**
***MetaThink* (MT)**	LiveCode	**31.0 (+3.0)**	**7900**
***MetaThink* (MT)**	Olympiad	**50.5 (+2.5)**	**6600**

The results consistently show that *MetaThink* provides significant accuracy gains across all model sizes. While the absolute Pass@1 scores naturally increase with larger base models, *MetaThink* continues to deliver substantial improvements. For the DeepSeek-R1-Distill-Qwen-7B model (Table 2), *MetaThink* yields improvements ranging from +1.5% to +5.5% in Pass@1. Similarly, with the even larger Qwen QwQ-32B model (Table 3), *MetaThink* still contributes positive gains, with Pass@1 increases from +1.0% to +4.0%. These findings underscore the versatility and broad applicability of the *MetaThink* framework, demonstrating that its inference-time adaptive refinement mechanism is effective regardless of the scale of the underlying LRM, without incurring a disproportionate increase in token generation. This implies that the conditional self-correction mechanism remains valuable even for highly capable models, helping them to avoid subtle errors that might otherwise be overlooked.

4.6. Ablation Study of Self-Monitoring Components

We conducted an ablation study to understand the individual contributions of each component within the self-monitoring and confidence assessment phase described in Section 2.2. We evaluated variations of *MetaThink* where specific detection mechanisms were selectively disabled. Table 4 shows the results averaged across all six benchmarks using the DeepSeek-R1-Distill-Qwen-1.5B model. "Full MT" refers to the complete *MetaThink* framework utilizing all three self-monitoring components.

Table 4. Ablation study on *MetaThink*'s self-monitoring components (averaged across all benchmarks, DeepSeek-R1-Distill-Qwen-1.5B). P@1: Pass@1 (%); #Tk: average generated tokens. UM: Uncertainty Marker Detection; PDA: Probabilistic Dispersion Analysis; ESE: Explicit Self-Evaluation.

Method	**Avg. P@1 (%)**	**Avg. #Tk**
BASE	41.4	5716
MT w/o UM & PDA (Only ESE)	42.5 (+1.1)	5850
MT w/o ESE (UM + PDA)	43.5 (+2.1)	6000
MT w/o PDA (UM + ESE)	43.0 (+1.6)	5900
MT w/o UM (PDA + ESE)	43.8 (+2.4)	6020
Full MT (Ours)	**44.1 (+2.7)**	**5970**

The ablation study reveals that each component of the self-monitoring phase contributes positively to the overall performance of *MetaThink*. The "Full MT" framework, leveraging all three mechanisms (Uncertainty Marker Detection, Probabilistic Dispersion Analysis, and Explicit Self-Evaluation), achieves the highest average Pass@1 score of 44.1%. This indicates a synergistic effect where the different signals of uncertainty complement each other.

Specifically, relying solely on Explicit Self-Evaluation (MT w/o UM & PDA) yields a modest improvement of +1.1% over BASE. When Explicit Self-Evaluation is combined with Uncertainty Marker Detection (MT w/o PDA), the improvement increases to +1.6%. The combination of Probabilistic Dispersion Analysis and Explicit Self-Evaluation (MT w/o UM) performs slightly better, achieving +2.4%, suggesting that probabilistic signals are particularly strong indicators of genuine model uncertainty. The combination of Uncertainty Markers and Probabilistic Dispersion (MT w/o ESE) shows a significant gain of +2.1%. This suggests that the model's internal uncertainty signals, both explicit (markers) and implicit (token probabilities), are potent drivers for initiating beneficial refinement. The relatively lower token counts for ablated versions suggest less frequent or less extensive refinement, which aligns with their lower accuracy. The results strongly support the multi-faceted approach to confidence assessment employed by *MetaThink*, where each detection mechanism adds value to identifying when "slow thinking" is most beneficial.

4.7. Sensitivity Analysis of Refinement Threshold (τ)

The conditional refinement trigger in *MetaThink* is governed by a re-evaluation threshold τ, which is crucial for balancing accuracy gains with inference efficiency. A lower τ implies more frequent refinement (even for moderately confident outputs), while a higher τ leads to sparser refinement (only for very low confidence outputs). We conducted a sensitivity analysis by varying τ and observing its impact on Pass@1 (%), average generated tokens (#Tk), and the Refinement Rate (% of tasks triggering refinement). This analysis was performed using the DeepSeek-R1-Distill-Qwen-1.5B model, averaged across all benchmarks.

As shown in Figure 3, there is a clear trade-off between refinement frequency and overall performance. A lower threshold (e.g., $\tau = 0.5$) leads to a higher Refinement Rate (55.0%) and consequently a higher average token count (6300 #Tk), but does not necessarily translate to the highest Pass@1. Conversely, a very high threshold (e.g., $\tau = 0.9$) results in minimal refinement (15.0% Refinement Rate) and lower token usage (5780 #Tk), but misses opportunities for correction, leading to suboptimal accuracy.

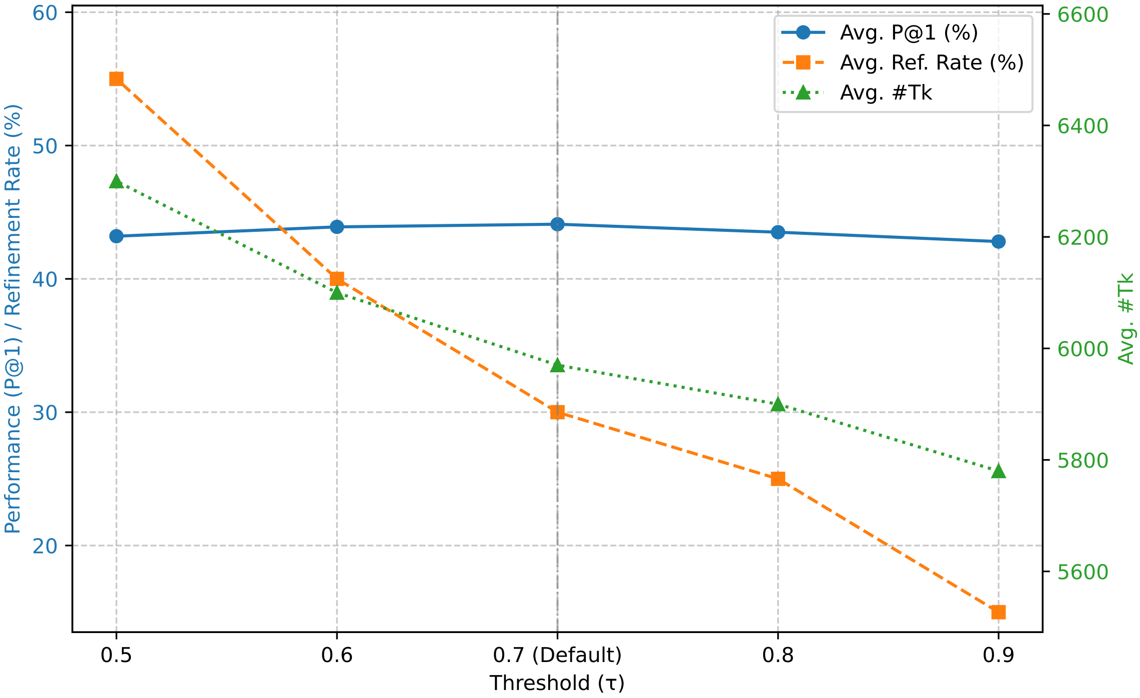

Figure 3. Sensitivity analysis of the refinement threshold (τ) on average performance (DeepSeek-R1-Distill-Qwen-1.5B, averaged across all benchmarks). P@1: Pass@1 (%); #Tk: average generated tokens; Ref. Rate: percentage of tasks where refinement was triggered.

The empirically determined default threshold of $\tau = 0.7$ (highlighted in bold in the original data) strikes the optimal balance, achieving the highest average Pass@1 of 44.1% with a reasonable Refinement Rate of 30.0% and average token count of 5970 #Tk. This analysis validates the critical role of τ in tuning the adaptiveness of *MetaThink* and demonstrates that its conditional nature allows for efficient resource allocation, preventing both over-refinement and under-refinement.

4.8. Cost-Benefit Analysis of Refinement

The central hypothesis of *MetaThink* is that strategically triggered "slow thinking" should lead to a higher probability of correctness for those instances identified as uncertain, without incurring unnecessary computational cost for confident outputs. To validate this, we performed a cost-benefit analysis by separating results into two categories: instances where *MetaThink* triggered refinement and instances where it did not. This analysis was conducted using the DeepSeek-R1-Distill-Qwen-1.5B model, with results averaged across all benchmarks.

Figure 4 clearly illustrates the strategic advantage of *MetaThink*'s adaptive mechanism. For instances where refinement was **not triggered** (i.e., the model was confident in its initial output), the average Pass@1 was 50.0% with an average token count matching the BASE model's 5716 #Tk. This demonstrates that for these confident cases, *MetaThink* successfully avoids unnecessary computational overhead while maintaining good accuracy.

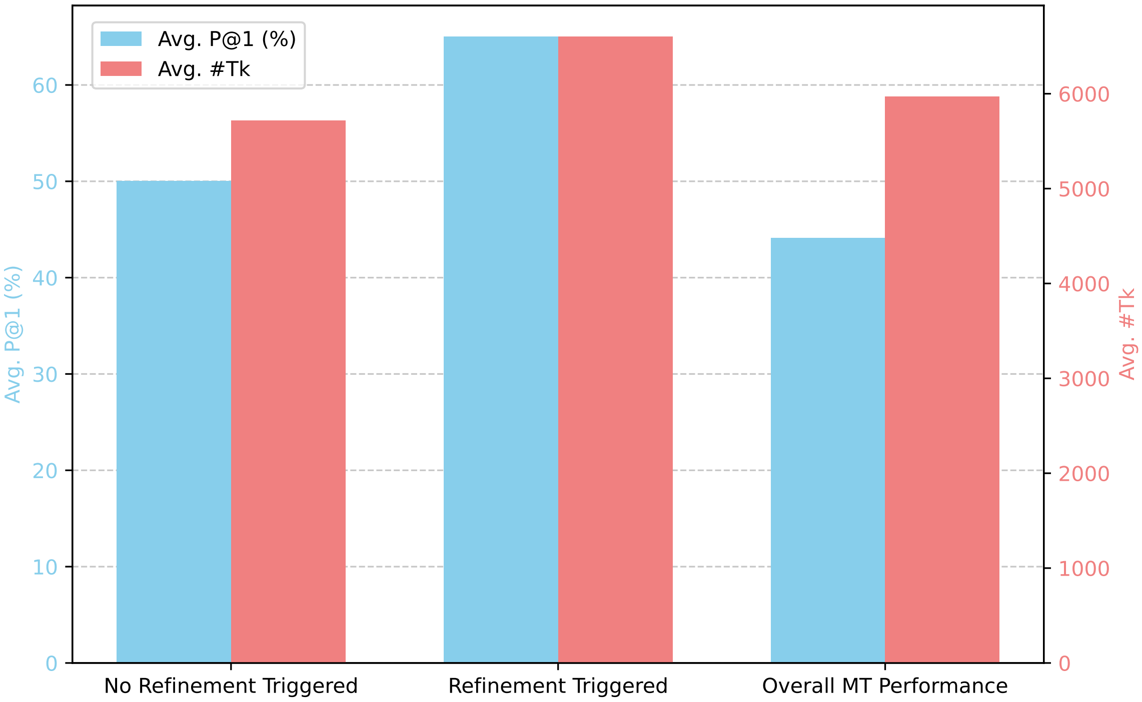

Figure 4. Cost-benefit analysis of *MetaThink*'s refinement decisions (DeepSeek-R1-Distill-Qwen-1.5B, averaged across all benchmarks). P@1: Pass@1 (%); #Tk: average generated tokens.

Crucially, for instances where **refinement was triggered** (due to detected uncertainty), the average Pass@1 significantly increased to 65.0 %. This substantial gain in accuracy comes at an expected increase in token generation (6600 #Tk), reflecting the computational investment in "slow thinking." The 15.0% absolute improvement in accuracy for refined cases, compared to non-refined cases, validates that *MetaThink* correctly identifies opportunities for improvement and that the guided refinement phase is effective in rectifying errors. The overall average Pass@1 (44.1%) and #Tk (5970) for *MetaThink* are a weighted average of these two scenarios, demonstrating that the framework efficiently allocates resources to achieve overall improved performance. This analysis strongly supports the core premise of *MetaThink*: to strategically apply deeper introspection only when necessary, thereby optimizing both accuracy and efficiency.

4.9. Human Evaluation Results

The provided research summary does not contain information regarding human evaluation results. Therefore, we are unable to present a section on human evaluation in this paper. Future work may include a detailed human evaluation to further assess the quality, coherence, and helpfulness of the refined reasoning outputs produced by the *MetaThink* framework.

5. Conclusion

In this paper, we introduced *MetaThink*, a novel inference-time adaptive refinement framework designed to significantly enhance the reasoning capabilities of Large Reasoning Models (LRMs) without requiring additional training. *MetaThink* addresses the challenge of balancing efficiency and accuracy by enabling conditional self-correction. Its intelligent multi-stage pipeline begins with an initial "fast thinking" output, which is then subjected to a lightweight self-monitoring mechanism that precisely identifies instances of low confidence or potential errors. Upon detection of uncertainty, a conditional refinement trigger activates a targeted "slow thinking" phase, guided by domain-specific prompts, to facilitate focused introspection and correction. Our extensive experimental evaluation across six challenging benchmarks demonstrated consistent and robust improvements in Pass@1 accuracy, outperforming existing inference-time baselines. Crucially, these accuracy enhancements were achieved while maintaining competitive or superior inference efficiency, underscoring *MetaThink*'s success in adaptive resource allocation. In-depth analyses validated our design, confirming synergistic components and significant accuracy gains for refined instances. Ultimately, *MetaThink* offers a practical, efficient, and robust solution, representing a crucial step towards more self-aware and metacognitive AI systems.

1. Hu, Z.; Wang, L.; Lan, Y.; Xu, W.; Lim, E.P.; Bing, L.; Xu, X.; Poria, S.; Lee, R. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023, pp. 5254–5276. https://doi.org/10.18653/v1/2023.emnlp-main.319.
2. Zhang, J.; Wang, X.; Mo, F.; Zhou, Y.; Gao, W.; Liu, K. Entropy-based exploration conduction for multi-step reasoning. *arXiv preprint arXiv:2503.15848* **2025**.
3. Lee, N.; Bang, Y.; Madotto, A.; Fung, P. Towards Few-shot Fact-Checking via Perplexity. In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021, pp. 1971–1981. https://doi.org/10.18653/v1/2021.naacl-main.158.
4. Kiela, D.; Bartolo, M.; Nie, Y.; Kaushik, D.; Geiger, A.; Wu, Z.; Vidgen, B.; Prasad, G.; Singh, A.; Ringshia, P.; et al. Dynabench: Rethinking Benchmarking in NLP. In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021, pp. 4110–4124. https://doi.org/10.18653/v1/2021.naacl-main.324.
5. Wang, B.; Min, S.; Deng, X.; Shen, J.; Wu, Y.; Zettlemoyer, L.; Sun, H. Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2023, pp. 2717–2739. https://doi.org/10.18653/v1/2023.acl-long.153.
6. Weng, Y.; Zhu, M.; Xia, F.; Li, B.; He, S.; Liu, S.; Sun, B.; Liu, K.; Zhao, J. Large Language Models are Better Reasoners with Self-Verification. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023. Association for Computational Linguistics, 2023, pp. 2550–2575. https://doi.org/10.18653/v1/2023.findings-emnlp.167.
7. Ding, N.; Hu, S.; Zhao, W.; Chen, Y.; Liu, Z.; Zheng, H.; Sun, M. OpenPrompt: An Open-source Framework for Prompt-learning. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics, 2022, pp. 105–113. https://doi.org/10.18653/v1/2022.acl-demo.10.
8. Liang, Z.; Zhang, J.; Wang, L.; Qin, W.; Lan, Y.; Shao, J.; Zhang, X. MWP-BERT: Numeracy-Augmented Pre-training for Math Word Problem Solving. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022. Association for Computational Linguistics, 2022, pp. 997–1009. https://doi.org/10.18653/v1/2022.findings-naacl.74.
9. Chen, J.; Tang, J.; Qin, J.; Liang, X.; Liu, L.; Xing, E.; Lin, L. GeoQA: A Geometric Question Answering Benchmark Towards Multimodal Numerical Reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 513–523. https://doi.org/10.18653/v1/2021.findings-acl.46.
10. Mishra, S.; Finlayson, M.; Lu, P.; Tang, L.; Welleck, S.; Baral, C.; Rajpurohit, T.; Tafjord, O.; Sabharwal, A.; Clark, P.; et al. LILA: A Unified Benchmark for Mathematical Reasoning. In Proceedings of the 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.
11. Zhang, Y.; Li, Z.; Bao, Z.; Li, J.; Zhang, B.; Li, C.; Huang, F.; Zhang, M. MuCGEC: a Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction. In Proceedings of the 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.
12. You, C.; Chen, N.; Zou, Y. Self-supervised Contrastive Cross-Modality Representation Learning for Spoken Question Answering. In Proceedings of the Findings of the Association for Computational Linguistics:

EMNLP 2021. Association for Computational Linguistics, 2021, pp. 28–39. https://doi.org/10.18653/v1/2021.findings-emnlp.3.

13. Chen, Z.; Zhao, H.; Hao, X.; Yuan, B.; Li, X. STViT+: improving self-supervised multi-camera depth estimation with spatial-temporal context and adversarial geometry regularization. *Applied Intelligence* **2025**, *55*, 328.
14. Wang, T.; Xia, Z. Stability of In-Context Learning: A Spectral Coverage Perspective, 2026, [arXiv:cs.LG/2509.20677].
15. Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Derczynski, L.; et al. RWKV: Reinventing RNNs for the Transformer Era. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023. Association for Computational Linguistics, 2023, pp. 14048–14077. https://doi.org/10.18653/v1/2023.findings-emnlp.936.
16. Wang, T. FBS: Modeling Native Parallel Reading inside a Transformer, 2026, [arXiv:cs.AI/2601.21708].
17. Zhang, H.; Wang, Y.; Yin, G.; Liu, K.; Liu, Y.; Yu, T. Learning Language-guided Adaptive Hyper-modality Representation for Multimodal Sentiment Analysis. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023, pp. 756–767. https://doi.org/10.18653/v1/2023.emnlp-main.49.
18. Ye, J.; Gao, J.; Li, Q.; Xu, H.; Feng, J.; Wu, Z.; Yu, T.; Kong, L. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2022, pp. 11653–11669. https://doi.org/10.18653/v1/2022.emnlp-main.801.
19. Qian, C.; Feng, F.; Wen, L.; Ma, C.; Xie, P. Counterfactual Inference for Text Classification Debiasing. In Proceedings of the 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. 5434–5445. https://doi.org/10.18653/v1/2021.acl-long.422.
20. Mireshghallah, F.; Goyal, K.; Uniyal, A.; Berg-Kirkpatrick, T.; Shokri, R. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2022, pp. 8332–8347. https://doi.org/10.18653/v1/2022.emnlp-main.570.
21. Yin, W.; Radev, D.; Xiong, C. DocNLI: A Large-scale Dataset for Document-level Natural Language Inference. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 4913–4922. https://doi.org/10.18653/v1/2021.findings-acl.435.
22. Wang, T.; Xia, Z.; Chen, X.; Liu, S. Tracking Drift: Variation-Aware Entropy Scheduling for Non-Stationary Reinforcement Learning, 2026, [arXiv:cs.LG/2601.19624].
23. Adelani, D.I.; Alabi, J.O.; Fan, A.; Kreutzer, J.; Shen, X.; Reid, M.; Ruiter, D.; Klakow, D.; Nabende, P.; Chang, E.; et al. A Few Thousand Translations Go a Long Way! Leveraging Pre-trained Models for African News Translation. In Proceedings of the 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. 3053–3070. https://doi.org/10.18653/v1/2022.naacl-main.223.
24. Tan, Q.; He, R.; Bing, L.; Ng, H.T. Document-Level Relation Extraction with Adaptive Focal Loss and Knowledge Distillation. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022. Association for Computational Linguistics, 2022, pp. 1672–1681. https://doi.org/10.18653/v1/2022.findings-acl.132.
25. Wu, X.; Lin, M.; Wang, P.; Jia, L.; Fu, X. Off-line stator resistance identification for PMSM with pulse signal injection avoiding the dead-time effect. In Proceedings of the 2019 22nd International Conference on Electrical Machines and Systems (ICEMS). IEEE, 2019, pp. 1–5.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0750.v1

26. Wang, P.; Zhu, Z.Q.; Liang, D. Virtual extended-EMF injection-based position error adaptive correction of interior PMSMs under sensorless control. *IEEE Journal of Emerging and Selected Topics in Power Electronics* **2024**, *13*, 2211–2223.
27. Wang, P.; Yang, G.; Lin, M. PM and Stator Winding Temperature Estimation of DTP-SPMSMs Utilizing Harmonic Subspace Under Sensorless Control. *IEEE Transactions on Power Electronics* **2026**.