

A Comprehensive Evaluation of LLM Reasoning: From Single-Model to Multi-Agent Paradigms

Yapeng Li¹, Jiakuo Yu¹, Zhixin Liu¹, Xinnan Liu¹, Jing Yu¹, Songze Li¹, Tonghua Su^{1*}

¹ Harbin Institute of Technology

{liyapeng, yujiakuo, Zhixin_Liu, liuxinnan, yujing, lisongze}@stu.hit.edu.cn, {thsu}@hit.edu.cn

Abstract

*Large Language Models (LLMs) are increasingly deployed as reasoning systems, where reasoning paradigms—such as Chain-of-Thought (CoT) and multi-agent systems (MAS)—play a critical role, yet their relative effectiveness and cost–accuracy trade-offs remain poorly understood. In this work, we conduct a comprehensive and unified evaluation of reasoning paradigms, spanning direct single-model generation, CoT-augmented single-model reasoning, and representative MAS workflows, characterizing their reasoning performance across a diverse suite of closed-form benchmarks. Beyond overall performance, we probe role-specific capability demands in MAS using targeted role isolation analyses, and analyze cost–accuracy trade-offs to identify which MAS workflows offer a favorable balance between cost and accuracy, and which incur prohibitive overhead for marginal gains. We further introduce **MIMeBench**, a new open-ended benchmark that targets two foundational yet underexplored semantic capabilities—semantic abstraction and contrastive discrimination—thereby providing an alternative evaluation axis beyond closed-form accuracy and enabling fine-grained assessment of semantic competence that is difficult to capture with existing benchmarks. Our results show that increased structural complexity does not consistently lead to improved reasoning performance, with its benefits being highly dependent on the properties and suitability of the reasoning paradigm itself. The codes are released at <https://gitcode.com/HIT1920/OpenLLMBench>.*

1. Introduction

Large Language Models (LLMs) [12, 21, 31] have become a foundational paradigm for general-purpose intelligence, demonstrating strong capabilities in complex reason-

ing, code synthesis, and scientific problem solving. Increasingly, LLMs are instantiated as *reasoning systems* that employ LLMs as core components for structured inference and decision-making, for which reliable operation under practical constraints such as accuracy, budget, and controllability becomes critical.

Within such *reasoning systems*, overall reasoning performance is no longer determined solely by model scale or training data, but increasingly depends on the *reasoning paradigm employed during inference*. Beyond direct single-pass generation, techniques such as CoT reasoning [29] and structured MAS workflows have been widely adopted as system-level paradigms to enhance reasoning quality. In practice, modern reasoning systems often combine multiple paradigms—for example, enabling CoT to strengthen a single model’s step-by-step reasoning, while further leveraging MAS workflows to mitigate CoT errors through external interaction and mutual critique.

Despite this progress, a key unresolved problem persists: the field still lacks a comprehensive and understanding of the cost–accuracy trade-offs of existing reasoning paradigms, as well as how their effectiveness varies across diverse scenarios when deployed as reasoning systems. Existing studies [4, 8, 20, 29] tend to focus on proposing new reasoning paradigms over a limited set of benchmarks, leaving several critical issues insufficiently characterized. In particular, it remains unclear under what circumstances CoT yields consistent accuracy gains rather than increased verbosity or output variance, and whether MAS provide benefits beyond a strong CoT-enabled single model or instead introduces additional instability. Moreover, comparisons among these paradigms under realistic budget constraints—where token consumption and multi-call overhead are important considerations—are still lacking.

Motivated by these gaps, we conduct a comprehensive study of reasoning paradigms from *single-model* to *multi-agent*, using an open-weight model¹ from the Pangu fam-

*Corresponding author

¹OpenPangu-Embedded-7B-V1.1. <https://ai.gitcode.com/ascend-tribe/openpangu-embedded-7b-model>

ily [1] as a representative instance. Concretely, we first establish a rigorous baseline by comparing direct single-model generation against its CoT-enabled counterpart, characterizing CoT’s precise impact on correctness and output stability. Building upon this, we systematically evaluate several representative MAS workflows—across a diverse suite of closed-form benchmarks, allowing us to map their effectiveness to specific task domains. We then investigate the interplay between these paradigms by assessing the performance of CoT-augmented MAS, examining whether internal deliberation and external collaboration yield synergistic or diminishing returns. Furthermore, we employ a role isolation protocol to probe the distinct capability demands imposed by different agent roles. Finally, our study concludes with a fine-grained, cost-aware analysis of evaluated MAS workflows, providing a clear characterization of the accuracy–cost trade-offs to identify which workflows offer a favorable balance of efficiency and reliability, and which incur prohibitive overhead for marginal gains. However, since our study relies on established closed-form benchmarks, such evaluations are limited to final-answer correctness. To address this limitation, we introduce **MIMeBench**, a new open-ended benchmark for main-idea multiple-choice option generation that directly probes two foundational semantic capabilities: *semantic abstraction* and *contrastive discrimination*. This provides a diagnostic view of the reasoning quality underlying the paradigms we study. Fig. 1 illustrates the overall structure of our study.

Our contributions are summarized as follows:

- We provide a comprehensive evaluation of reasoning paradigms spanning direct generation, CoT-enabled single-model reasoning, and representative MAS workflows, measuring performance under a unified framework.
- We introduce MIMeBench, a new open-ended benchmark designed to assess semantic abstraction and contrastive discrimination ability. MIMeBench provides an additional evaluation axis by directly measuring the foundational semantic skills required for robust reasoning.
- We conduct a detailed analysis of several MAS workflows by examining role-specific capability demands, and by analyzing cost–accuracy trade-offs to determine which workflows offer a favorable accuracy–cost balance and which exhibit diminishing returns.

2. Related Work

2.1. Benchmarks for LLMs

Benchmarks play a central role in evaluating and comparing large language models, serving as the primary basis for assessing progress across reasoning, knowledge, and code generation.

Existing benchmarks differ substantially in both task formulation and evaluation strategy, and can be broadly grouped into *Closed-Form* Benchmarks, where model outputs are assessed against well-defined ground-truth answers, and *Open-Ended* Benchmarks, where evaluation requires more open-ended judgment.

Closed-Form Benchmarks. Closed-form benchmarks span multiple task domains—such as mathematical reasoning, general understanding, and code generation—where they evaluate model outputs using exact answers or deterministic verification procedures. In mathematical reasoning domain, GSM8K [6] serves as a foundational benchmark for grade-school level problems, while AQUA [17] targets numerical and algebraic reasoning over text in a multiple-choice setting, and GSM-Hard [10] together with competition-level datasets such as AIME-2024 increase difficulty while preserving answer determinacy. In general understanding domain, ARC [5] comprises grade-school science questions with Easy and Challenge splits; CommonsenseQA [25] targets commonsense knowledge questions; GPQA-Diamond [22] is an expert-written 198-question subset spanning biology, chemistry, and physics—all these benchmarks are multiple-choice, with a single correct option as ground truth. In code generation domain, HumanEval [3] adopts a closed-form paradigm by judging functional correctness through unit tests, later strengthened by HumanEval+ [18] with expanded test coverage to improve reliability and reduce false positives.

Open-Ended Benchmarks. Open-ended benchmarks target generative tasks where model outputs cannot be evaluated against a single canonical answer, and thus rely more heavily on evaluation procedures. Traditional automatic metrics such as BLEU [15] and ROUGE [16] offer scalable scoring but are limited to surface-level overlap and fail to capture semantic correctness or reasoning quality. To address these limitations, recent benchmarks adopt large language models as judges for open-ended evaluation. MT-Bench [32] reports strong agreement between LLM-based judgments and human evaluations. Building on this paradigm, GPTScore [9] and G-Eval [19] further formalize evaluation through multi-dimensional criteria and explicit reasoning, improving the reliability and interpretability of open-ended benchmark assessment.

2.2. LLM-Based Multi-Agent Systems

Recent advances in LLM reasoning increasingly emphasize structured inference workflows, aiming to improve performance and reliability beyond single-path generation. Early approaches such as self-ensemble methods [28, 30] and iterative self-refinement frameworks [20, 23, 24] embody this perspective within a single-model setting, by encouraging a

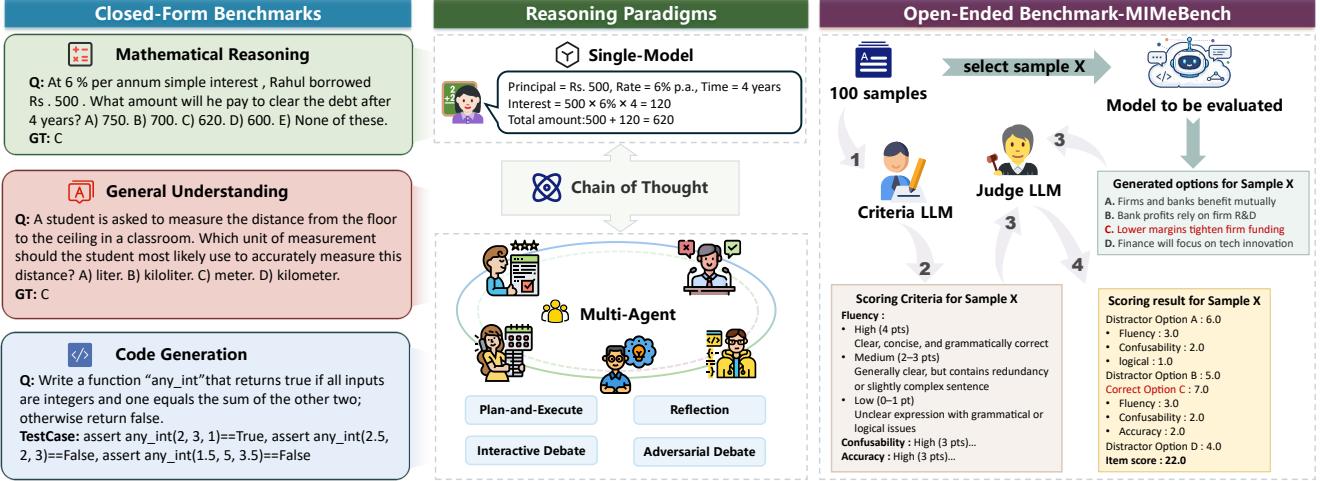


Figure 1. **Overview of our study.** We evaluate multiple reasoning paradigms under a unified protocol using closed-form benchmarks (**left**), and complement them with an open-ended benchmark, MIMeBench (**right**).

single model to generate and aggregate multiple reasoning trajectories or to iteratively revise its outputs through internal feedback. While effective in improving accuracy, these methods are inherently constrained by single-model introspection and limited exploration, and may degrade when initial reasoning becomes overly confident.

Building upon this workflow-centric perspective, subsequent work generalizes these ideas by externalizing reasoning, critique, and aggregation into explicit interactions among multiple agents. This line of work gives rise to *MAS*, in which distinct agents are explicitly assigned to different roles or stages of the inference workflow, jointly carrying out complex reasoning through coordinated inter-agent interactions [13, 27]. Representative multi-agent debate frameworks [8, 14] show that exchanging conflicting viewpoints can encourage divergent reasoning and improve performance on complex and counter-intuitive tasks. Extensions such as RECONCILE [2] and multi-agent verification [15] further highlight the importance of agent diversity, consensus mechanisms, and verification in improving reasoning quality and decision reliability.

In addition, some MAS frameworks move beyond loosely coupled agent interactions and explicitly formalize reasoning workflows as structured, role-based or stage-based decompositions. Systems such as MetaGPT [11] and AgentVerse [4] decompose complex tasks into coordinated phases, such as planning, execution, and evaluation—enabling fine-grained control and coordination in multi-step problem solving.

3. Preliminary

This section introduces the notations and formalizes the difference between *single-model* and *multi-agent reasoning*

paradigms. We also define the MAS workflows evaluated in this work, which are constructed based on prior work.

3.1. Notation

Let x denote the task input (e.g., a question, a problem statement, or a coding prompt), and y denote the final output (e.g., an answer or a code solution). We use \mathcal{M}_θ to denote an LLM with parameters θ . A decoding procedure induces a conditional distribution:

$$p_\theta(y | x) \triangleq \mathcal{M}_\theta(x). \quad (1)$$

For any intermediate text artifact (e.g., a plan, critique or an explicit CoT procedure), we denote it by z . A general reasoning process can be viewed as producing a sequence of intermediate artifacts $\mathbf{z} = (z_1, z_2, \dots, z_T)$ and then the final output y :

$$p_\theta(y, \mathbf{z} | x) = \prod_{t=1}^T p_\theta(z_t | x, z_{<t}) \cdot p_\theta(y | x, \mathbf{z}). \quad (2)$$

We use $\mathcal{C}(\cdot)$ to denote inference cost (token consumption). For a dialog-style workflow producing messages $\{m_k\}_{k=1}^K$, we write

$$\mathcal{C} \triangleq \sum_{k=1}^K |m_k|. \quad (3)$$

where $|m_k|$ is the number of tokens in message m_k .

3.2. Single-Model Reasoning Paradigm

We first formalize the *single-model reasoning paradigm*, where a single model instance produces the final answer in one pass:

$$y = f_\theta(x), \quad \text{where } f_\theta(x) \sim p_\theta(y | x). \quad (4)$$

Optionally, single-model reasoning may generate an explicit CoT procedure z :

$$z \sim p_\theta(z | x), \quad y \sim p_\theta(y | x, z). \quad (5)$$

In practice, Pangu-7B supports two inference strategies that can be abstracted as: (i) **Direct Response** (no_think): $y \sim p_\theta(y | x)$, (ii) **Adaptive Reasoning** (auto_think): $y \sim p_\theta(y | x, z)$ with z generated adaptively.

Accordingly, the inference cost is dominated by a single forward generation, with an optional CoT procedure z :

$$y \sim p_\theta(y | x, z), \quad \mathcal{C} \approx O(|y| + |z|). \quad (6)$$

where $|z| = 0$ in the direct-response setting.

3.3. Multi-Agent Reasoning Paradigm

MAS externalize reasoning into explicit interactions among multiple agent instances. Let there be N agents $\{\mathcal{A}_i\}_{i=1}^N$, where each agent \mathcal{A}_i is an instantiation of (possibly the same) base model \mathcal{M} under a role-specific prompt π_i :

$$\mathcal{A}_i(\cdot) \triangleq \mathcal{M}(\pi_i, \cdot). \quad (7)$$

A general MAS workflow defines (1) a message-passing protocol and (2) a termination rule producing the final output:

$$m_k = g_k(x, m_{<k}), \quad y = h(x, m_{1:K}). \quad (8)$$

where g_k specifies which agent speaks at step k and what context it receives, and h aggregates the transcript to form the final prediction.

Compared to single-model, *multi-agent reasoning paradigm* introduces explicit interactive messages, and its inference cost scales with the total length of all generated messages:

$$\mathcal{C} \approx O\left(\sum_k |m_k| + |y|\right). \quad (9)$$

3.4. MAS Workflows

As illustrated in Fig. 2, we formalize four MAS workflows evaluated in this work—*Plan-and-Execute*, *Reflection*, *Interactive Debate*, and *Adversarial Debate*.

(1) Plan-and-Execute. This workflow decomposes problem solving into planning and execution using two agents: a Planner $\mathcal{A}_{\text{plan}}$ and an Executor $\mathcal{A}_{\text{exec}}$. First, the Planner generates a plan z_{plan} :

$$z_{\text{plan}} \sim p_{\text{plan}}(z | x), \quad (10)$$

then the Executor produces the final answer conditioned on the plan:

$$y \sim p_{\text{exec}}(y | x, z_{\text{plan}}). \quad (11)$$

This design isolates strategic decomposition from instruction-following fidelity.

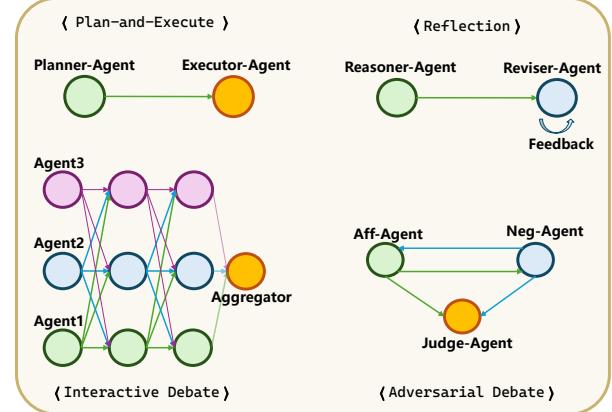


Figure 2. Overview of MAS Workflows.

(2) Reflection. This workflow performs iterative correction with two phases. First, a Reasoner \mathcal{A}_{rsn} generates an initial solution $y^{(0)}$ (and its rationale). Then, a Reviser \mathcal{A}_{rev} first produces an explicit feedback artifact z_{fee} by critiquing the initial solution, and subsequently generates a revised solution $y^{(1)}$ conditioned on this feedback:

$$y^{(0)} \sim p_{\text{rsn}}(y | x), \quad (12)$$

$$z_{\text{fee}} \sim p_{\text{rev}}(z | x, y^{(0)}), \quad (13)$$

$$y^{(1)} \sim p_{\text{rev}}(y | x, y^{(0)}, z_{\text{fee}}). \quad (14)$$

We take $y \triangleq y^{(1)}$ as the final output.

(3) Interactive Debate. Let there be N peer debaters $\{\mathcal{A}_i\}_{i=1}^N$ and an Aggregator \mathcal{A}_{agg} . Each debater first produces an independent solution:

$$y_i^{(0)} \sim p_i(y | x), \quad i = 1, \dots, N. \quad (15)$$

For debate rounds $r = 1, \dots, R$, each agent updates its answer conditioned on other agents' synthesized messages $\text{Sync}(\cdot)$:

$$y_i^{(r)} \sim p_i(y | x, \text{Sync}(y_{-i}^{(r-1)})), \quad (16)$$

where $y_{-i}^{(r-1)}$ denotes the set of other agents' solutions at round $r - 1$. Finally, \mathcal{A}_{agg} produces the final output by examining all candidate answers $\{y_i^{(R)}\}_{i=1}^N$ and selecting the most frequently occurring one:

$$y = \mathcal{A}_{\text{agg}}(\{y_i^{(R)}\}_{i=1}^N). \quad (17)$$

(4) Adversarial Debate. This workflow assigns explicit opposing roles: an Affirmative agent \mathcal{A}_{aff} , a Negative agent \mathcal{A}_{neg} , and a Judge $\mathcal{A}_{\text{judge}}$. The Affirmative proposes an initial solution $y_{\text{aff}}^{(0)}$, and the Negative responds with a counter-

solution $y_{\text{neg}}^{(0)}$:

$$y_{\text{aff}}^{(0)} \sim p_{\text{aff}}(y | x), \quad (18)$$

$$y_{\text{neg}}^{(0)} \sim p_{\text{neg}}(y | x, y_{\text{aff}}^{(0)}). \quad (19)$$

For rebuttal rounds $r = 1, \dots, R$, each side responds to the opponent’s latest message:

$$y_{\text{aff}}^{(r)} \sim p_{\text{aff}}(y | x, y_{\text{neg}}^{(r-1)}), \quad (20)$$

$$y_{\text{neg}}^{(r)} \sim p_{\text{neg}}(y | x, y_{\text{aff}}^{(r)}). \quad (21)$$

The Judge $\mathcal{A}_{\text{judge}}$ then outputs the final decision after reading the complete debate transcript \mathcal{T} :

$$\mathcal{T} = \{y_{\text{aff}}^{(r)}, y_{\text{neg}}^{(r)}\}_{r=0}^R, \quad (22)$$

$$y \sim p_{\text{judge}}(y | x, \mathcal{T}). \quad (23)$$

4. MIMeBench

We introduce **MIMeBench**, a benchmark for *Main-Idea Multiple-Choice Question (MCQ) Generation*, to evaluate foundational semantic skills that underpin effective reasoning. Unlike closed-form benchmarks, which primarily assess final-answer correctness, MIMeBench directly evaluates a model’s ability to (i) identify the core semantics of a passage and (ii) distinguish between semantically similar yet meaningfully distinct alternatives. These capabilities correspond to *semantic abstraction* and *contrastive discrimination*, respectively.

Rather than assessing only whether a model produces a correct final answer, this open-ended formulation directly measures the quality of two foundational reasoning components—*semantic abstraction* and *contrastive discrimination*—by evaluating how accurately core meaning is extracted and how effectively semantically challenging alternatives are constructed, thereby yielding interpretable signals that help explain and predict performance on complex reasoning tasks.

This section describes the construction of MIMeBench, its dynamic, item-specific evaluation criteria, and the scoring and aggregation protocol used for model assessment.

4.1. Dataset Construction

The dataset is compiled from official National Civil Service Examination items and multiple provincial Administrative Aptitude Test (AAT) exams collected over the past five years. We select 100 main-idea summarization samples covering diverse topics and discourse structures. Each item is derived from a real examination question and consists of a passage, a question (typically phrased as “*This passage is intended to illustrate...* ”), and four *expert-designed* options as reference, including one correct main-idea option and three distractors.

Given the passage and prompt, a model is required to generate four new options following the same structure—one correct option and three distractors—mirroring the format and difficulty of authentic examination items. Passage length and difficulty are controlled to reduce bias from extreme cases, while topic diversity is maintained to evaluate contextual generalization. For exam security and compliance reasons, we do not release the original items or full passages.

4.2. Dynamic Evaluation Criteria

Unlike closed-form benchmarks with fixed answers or static rubrics, MIMeBench relies on *item-specific evaluation criteria* that capture the semantic structure and distractor logic of each item. This design is motivated by the observation that each item differ substantially in discourse organization, thematic focus, and plausible distractor strategies, making a single global rubric inadequate.

For each benchmark item, a criteria model, denoted as M_{crit} , is prompted to analyze the source passage together with the original reference options from the item. By using these *expert-designed* options, M_{crit} can derive criteria that align with the experts’ intended interpretation and quality standards for the item. Accordingly, this model is used exclusively to generate item-specific evaluation criteria and is not involved in option generation or scoring. Based on these information, the model generates two sets of evaluation criteria: (i) criteria for assessing the correct option, and (ii) criteria for assessing distractor options. Within each item, the three distractors are evaluated against the *same* set of distractor criteria to enforce a uniform judging standard, ensuring that the resulting scores are directly comparable across distractors. To reduce stochasticity, three independent sets of criteria are generated for correct options and three for distractors, and scores obtained under these criteria are averaged during aggregation.

4.3. Scoring Dimensions and Aggregation

We formalize the scoring process using explicit notation. For a given item, let o^* denote the correct option generated by the evaluated model, and $\{o_1, o_2, o_3\}$ denote the three generated distractors. Let $\mathcal{C}^* = \{c_1^*, c_2^*, c_3^*\}$ denote the three independently generated evaluation criteria for the correct option, and $\mathcal{C}^- = \{c_1^-, c_2^-, c_3^-\}$ denote the three evaluation criteria shared by all distractors.

Each correct option is evaluated along three dimensions *fluency*, *confusability*, and *accuracy*—and each distractor along *fluency*, *confusability*, and *logical consistency*, with the scores of the three dimensions summing to a total of 10 points per option. The weighting of these dimensions is fixed across all items. Here, *fluency* measures grammatical-ity and readability; *accuracy* measures whether the correct option captures the main idea; for correct options, *confus-*

ability rewards paraphrased expressions that are not trivially anchored by lexical overlap with the source passage(e.g., copying many words), whereas for distractors *confusability* measures how misleading the option is; *logical consistency* checks whether a distractor is internally coherent and not self-contradictory.

For the correct option, the aggregated score is computed as:

$$S^* = \frac{1}{|\mathcal{C}^*|} \sum_{k=1}^{|\mathcal{C}^*|} J(o^* | c_k^*), \quad (24)$$

where $J(\cdot | c)$ denotes the judge model scoring an option under criterion c .

Similarly, each distractor o_i is scored as:

$$S_i = \frac{1}{|\mathcal{C}^-|} \sum_{k=1}^{|\mathcal{C}^-|} J(o_i | c_k^-), \quad i \in \{1, 2, 3\}. \quad (25)$$

The final item-level score is:

$$S_{\text{item}} = S^* + \sum_{i=1}^3 S_i. \quad (26)$$

Model performance on MIMeBench is reported as the mean item score over the dataset. Algorithm 1 summarizes the full dataset-level evaluation pipeline.

5. Experiments

This section reports our experimental design and empirical findings. We first describe the evaluation setup, benchmarks, and scoring methodology (Sec. 5.1). We then present single-model inference results, including cross-model comparisons and the impact of inference strategies (Sec. 5.2), followed by multi-agent inference results under representative MAS workflows (Sec. 5.3). Finally, we report open-ended evaluation results on MIMeBench. (Sec. 5.4).

5.1. Experimental Protocol

5.1.1. Setup

We adopt Pangu-7B¹ model from the Pangu family, which is developed within the Ascend ecosystem. Accordingly, all our evaluation experiments are conducted in an Ascend-based environment, with the models deployed on Ascend 910B NPUs.

To ensure reproducibility and consistency, all evaluated models we used (not only Pangu-7B) are run with the default decoding hyperparameters specified in their open-source configurations (temperature, top_p, and top_k). The maximum context length is set to each model’s maximum supported embedding length. Except for MIMeBench, all benchmarks are conducted under a unified zero-shot setting without additional prompting or task-specific guidance.

Algorithm 1: MIMeBench evaluation pipeline.

Input: Dataset $\mathcal{D} = \{(p^{(n)}, q^{(n)}, \mathcal{R}^{(n)})\}_{n=1}^N$ ($N=100$), evaluated model M , criteria model M_{crit} , judge model J , criteria prompts π^* for correct-option criteria and π^- for distractor criteria

Output: Mean MIMeBench score \bar{S}

p: passage text; q: prompt used to elicit M to generate options; R: reference options;

Total $\leftarrow 0$;

for $n \leftarrow 1$ **to** N **do**

$(p, q, \mathcal{R}) \leftarrow (p^{(n)}, q^{(n)}, \mathcal{R}^{(n)})$;

$(o^*, o_1, o_2, o_3) \leftarrow M(p, q)$;

$\mathcal{C}^* \leftarrow \emptyset$;

for $k \leftarrow 1$ **to** 3 **do**

$c \leftarrow M_{\text{crit}}(p, \mathcal{R}; \pi^*)$;

$\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{c\}$;

$\mathcal{C}^- \leftarrow \emptyset$;

for $k \leftarrow 1$ **to** 3 **do**

$c \leftarrow M_{\text{crit}}(p, \mathcal{R}; \pi^-)$;

$\mathcal{C}^- \leftarrow \mathcal{C}^- \cup \{c\}$;

$S^{(n)} \leftarrow \text{ITEMSCORE}(o^*, o_1, o_2, o_3, \mathcal{C}^*, \mathcal{C}^-, J)$;

Total \leftarrow Total + $S^{(n)}$;

$\bar{S} \leftarrow \text{Total}/N$;

return \bar{S} ;

Function ITEMSCORE($o^*, o_1, o_2, o_3, \mathcal{C}^*, \mathcal{C}^-, J$):

$S^* \leftarrow \text{AVGCRITSCORE}(o^*, \mathcal{C}^*, J)$;

$S^- \leftarrow 0$;

for $i \leftarrow 1$ **to** 3 **do**

$S^- \leftarrow S^- + \text{AVGCRITSCORE}(o_i, \mathcal{C}^-, J)$;

return $S^* + S^-$;

Function AVGCRITSCORE(o, \mathcal{C}, J):

$s \leftarrow 0$;

foreach $c \in \mathcal{C}$ **do**

$s \leftarrow s + J(o | c)$;

return $s/|\mathcal{C}|$;

5.1.2. Benchmarks

To comprehensively evaluate the model’s capabilities under diverse reasoning demands, we adopt a suite of closed-form benchmarks, covering both standard evaluations and more rigorous variants. This suite enables a holistic assessment of exact reasoning performance and answer correctness. The specific tasks and their corresponding evaluation metrics are summarized in Table 1.

In addition to closed-form benchmarks, we include MIMeBench as an open-ended generation benchmark. The evaluated model is required to generate a set of options for a question, where quality is assessed by semantic adequacy and distractor plausibility rather than exact string matching. Following the protocol in Sec. 4, we use a LLM-based judge to score model outputs and report performance using mean

Table 1. Selected benchmarks in our work, covering mathematical reasoning, general understanding, and code generation domains (referred to as **Math**, **General**, and **Code** in later analyses), together with an open-ended generation benchmark—MIMeBench.

Domain	Datasets	Metric
Mathematical Reasoning	AQUA	Accuracy
	GSM8K	
	GSM-Hard	
	AIME-2024	
General Understanding	ARC-Easy	Accuracy
	ARC-Challenge	
	CommonsenseQA	
	GPQA-Diamond	
Code Generation	HumanEval	Pass@1
	HumanEval+	
Open-ended Generation	MIMeBench	Avg. Score

scores.

5.1.3. Evaluation Methodology

To maintain consistency and assessment fidelity for closed-form benchmarks (excluding MIMeBench), we adopt a zero-shot evaluation framework in which Qwen3-32B is used as an automated judge to compare model outputs against ground-truth answers. This framework mitigates parsing errors and standardizes the scoring methodology. We detail the evaluation procedures for different benchmarks below:

- **Math & General:** For non-coding benchmarks, the model’s output and ground truth are fed into Qwen3-32B. The judge performs approximate equivalence checking to ascertain correctness, yielding a binary score of 1 (Correct) or 0 (Incorrect).
- **Code:** For coding benchmarks, we primarily rely on a rule-based matching procedure to extract executable code blocks. In cases where the rule-based approach fails to produce a valid extraction, we fall back to using Qwen3-32B as an extractor to isolate the executable code blocks. The extracted blocks are then evaluated against a standard unit test suite: a sample is assigned a score of 1 (Pass) only if it passes all test cases; otherwise, it is assigned 0 (Fail).

For the automated judge, we set the decoding temperature to 0 to reduce stochasticity and promote fair and stable judgments.

5.2. Single-Model Inference Results

We first establish a model-grounded reference for reasoning performance to situate the subsequent analysis. Adhering to the protocols defined in Sec. 5.1, we assess Pangu-7B across the selected benchmarks. We benchmark Pangu-7B

against contemporary open-weight reasoning models, including the Qwen3 series [26] and the DeepSeek-R1 distilled variants [7], and also report its results under both direct-generation and thinking strategies. Together, these results delineate the empirical regime in which our later comparisons are made.

5.2.1. Comparison with State-of-the-Art Baselines

We benchmark Pangu-7B (`auto_think`) against Qwen3 (8B/14B) and DeepSeek-R1 (Distill-Llama-8B/Qwen3-8B). To ensure a fair comparison, all models are evaluated in their thinking modes.

Competitive Analysis. As illustrated in Table 2, while Qwen3 variants exhibit robust performance on standard benchmarks (GSM8K, ARC-Challenge), Pangu-7B differentiates itself through superior proficiency in high-difficulty reasoning tasks:

- **Math:** Pangu-7B attains an accuracy of 86.67% on AIME-24, surpassing both Qwen3-8B (80.00%) and the specialized DeepSeek-R1-Distill-Qwen3-8B (80.00%) by a substantial margin. This suggests enhanced robustness in handling competition-level mathematical problems.
- **Code:** On the more stringent HumanEval+ benchmark, Pangu-7B reaches 90.24%, outperforming Qwen3-14B (89.02%) and leading the 8B-class models significantly.
- **General:** In expert-level GPQA-Diamond, Pangu-7B (76.77%) exceeds its direct competitors Qwen3-8B (75.76%) and DeepSeek-R1 variants, trailing only the larger Qwen3-14B model.

These findings imply that Pangu-7B’s architecture trades marginal regressions in standard tasks for considerable gains in complex reasoning and synthesis capabilities, positioning it as a highly specialized model for demanding domains.

5.2.2. Impact of Inference strategies

We scrutinize the efficacy of two inference strategies defined in Sec. 3.2. Table 3 tabulates the comparative results between the direct response (`no_think`) and the adaptive reasoning (`auto_think`) strategy.

Data presented in Table 3 indicate that activating the `auto_think` mechanism confers consistent performance uplifts. Specifically, these gains are most pronounced in frontier-level tasks necessitating complex logic synthesis, such as AIME-2024 (+26.67%) and GPQA-Diamond (+8.08%). This validates that the CoT procedure effectively bridges the gap between intuitive retrieval and rigorous problem-solving.

5.3. Multi-Agent Inference Results

We evaluate Pangu-7B under MAS workflows in Sec. 3.4 and compare them against Single-Model Inference, with re-

Table 2. Comparison with state-of-the-art open-weight models.

Domain	Task	Metric	Pangu	Qwen3		DeepSeek-R1 Distill	
			7B	8B	14B	Llama-8B	Qwen3-8B
Math	GSM8K	Acc.	94.54	97.19	97.12	82.26	95.53
	AIME-2024	Acc.	86.67	80.00	80.00	53.33	80.00
General	ARC-Challenge	Acc.	90.02	96.33	95.48	88.14	95.73
	GPQA-Diamond	Acc.	76.77	75.76	79.29	54.04	67.68
Code	HumanEval+	Pass@1	90.24	88.41	89.02	83.54	87.80

Table 3. Performance comparison of Pangu-7B under the two inference strategies.

Domain	Task	Total	no_think		auto_think		
			Correct	Success Rate	Correct	Success Rate	Δ
Math	AQUA	254	223	87.80	230	90.55	+2.75
	GSM8K	1319	1234	93.56	1247	94.54	+0.98
	GSM-Hard	1319	814	61.71	869	65.88	+4.17
	AIME-2024	30	18	60.00	26	86.67	+26.67
General	ARC-Easy	2376	2233	93.98	2281	96.00	+2.02
	ARC-Challenge	1172	1018	86.86	1055	90.02	+3.16
	GPQA-Diamond	198	136	68.69	152	76.77	+8.08
Code	HumanEval	164	138	84.15	157	95.73	+11.58
	HumanEval+	164	130	79.27	148	90.24	+10.97

sults summarized in Table 4.

Under the no_think strategy, MAS workflows exhibit highly task-dependent effects. Reflection consistently improves performance across benchmarks, indicating strong self-correction capability, while Plan-and-Execute is particularly effective for structured tasks such as code generation. However, these gains come with clear trade-offs: strategies that benefit one task can impair others. For example, rigid Plan-and-Execution negatively impacts commonsense reasoning, and Adversarial Debate introduces substantial interference on tasks requiring precise, convergent logic. Overall, these results suggest that no single MAS design is universally optimal; effective collaboration patterns must be aligned with task characteristics.

We further examine MAS performance on top of (auto_think) strategy, as shown in Table 5. While auto_think substantially strengthens the baseline, additional multi-agent interactions provide limited and inconsistent benefits. In some cases, external debate complements internal reasoning, but in others, MAS integration leads to diminished performance. This pattern indicates diminishing returns during inference: once high-quality solutions are produced internally, additional agent interactions may introduce noise rather than useful evidence. This behavior is further illustrated through qualitative case studies in the

Table 4. MAS results under the no_think strategy. The delta (Δ) compares accuracy to the single-model inference baseline shown in Table 3. The best-performing framework for each task is highlighted in bold.

Task	Selected MAS	Success Rate	Δ
GSM-Hard	Plan-and-Execute	63.38	+1.67
	Interactive Debate	62.09	+0.38
	Reflection	67.78	+6.07
	Adversarial Debate	48.37	-13.34
ARC-Challenge	Plan-and-Execute	79.35	-7.51
	Interactive Debate	90.27	+3.41
	Reflection	91.81	+4.95
	Adversarial Debate	82.51	-4.35
HumanEval	Plan-and-Execute	92.68	+8.53
	Interactive Debate	85.37	+1.22
	Reflection	89.02	+4.87
	Adversarial Debate	78.66	-5.49

Appendix D.

5.4. Results on MIMeBench

To assess the foundational skills of **semantic abstraction** and **contrastive discrimination**, we evaluated several 7B-

Table 5. MAS results under the `auto_think` strategy. The delta (Δ) compares accuracy to the single-model inference baseline shown in Table 3.

Task	Selected MAS	Success Rate	Δ
GSM-Hard	Plan-and-Execute	64.52	-1.36
ARC-Challenge	Interactive Debate	91.55	+1.53
HumanEval	Reflection	91.46	-4.27

scale models on MIMeBench, including general-purpose baselines (e.g., Qwen2.5-7B², DeepSeek-7B³) and a non-publicly available Specialized MCQ Generator. This analysis moves beyond final-answer correctness to assess the quality of the reasoning components themselves: identifying a main idea (abstraction) and constructing plausible yet incorrect alternatives (discrimination).

The results in Table 6 reveal that proficiency in these foundational skills correlates with strong reasoning performance. While Pangu-7B underperforms the Specialized MCQ Generator, it demonstrates a clear advantage over other general-purpose baselines. This advantage is twofold:

- First, Pangu-7B attains the highest correct-option score, a direct measure of its superior **semantic abstraction** capability in extracting a passage’s central theme.
- Second, it generates the most effective distractors, evidenced by the highest mean distractor score. This indicates a stronger capacity for **contrastive discrimination**—the ability to create semantically challenging alternatives that test for true comprehension.

This direct evidence on foundational skills provides a compelling explanation for the robust performance Pangu-7B demonstrated on complex reasoning benchmarks (results in Sec. 5.2). A model that excels at identifying a problem’s core semantics and distinguishing between nuanced, competing hypotheses is inherently better equipped to execute a reliable step-by-step reasoning process. The strength observed here is not merely about generating plausible text, but about the underlying *semantic precision* that makes complex reasoning possible.

6. MAS Analysis: Roles, Cost, and Accuracy

To complement the aggregate results reported in Section 5, this section presents additional analyses that go beyond end-to-end accuracy. Specifically, we examine role-specific capability demands in MAS and analyze the trade-offs between inference cost and accuracy across different workflows.

²Qwen2.5-7B-Instruct. <https://www.modelscope.cn/models/Qwen/Qwen2.5-7B-Instruct>

³DeepSeek-R1-Distill-Qwen-7B. <https://www.modelscope.cn/models/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B>

Table 6. Evaluation results on MIMeBench. **Avg.** denotes the average dataset-level score \bar{S} aggregated over all options; **Corr.** denotes the mean score S^* of the correct main-idea option; **Wrong.** denotes the mean score S^- of three distractor options. **Specialized MCQ Generator** refers to a closed-source model adapted for MCQ generation. All models are evaluated under the thinking strategy.

Model	Avg.	Corr.	Wrong.
Qwen2.5-7B	22.20	5.65	5.52
Pangu-7B	24.26	6.38	5.96
DeepSeek-7B	21.53	5.25	5.43
Specialized MCQ Generator	28.08	8.97	6.37

6.1. Role-Specific Capability Demand Analysis

To better understand the capability demands imposed by different agent roles, we analyze model outputs under role-isolated MAS workflows. Rather than focusing on the overall outcome of a MAS workflow, this analysis examines how individual roles—*planner*, *reviser*, and *aggregator*—differ in the types of reasoning competence they require from a model.

For each MAS, the collaborative context is fixed and the evaluated model is assigned to a single role at a time (see Appendix A for more details). This allows different models to be compared under identical role-specific inputs, isolating how effectively they satisfy the capability requirements of each role, independent of interaction effects.

As shown in Table 7, the capability demands of different roles vary substantially. Performance differences across models are relatively small for the *Planner* and *Aggregator* roles while the *Reviser* role exhibits much larger variance. Notably, Pangu-7B demonstrates a clear advantage in the *Reviser* role, achieving the highest revision accuracy among all compared models. This suggests that its strength lies in post-hoc reasoning behaviors, including critiquing partially correct solutions and producing focused improvements, rather than in planning or aggregation alone. Such results aligns with its strong performance under Reflection-based workflows observed in earlier experiments.

More broadly, this role-dependent pattern helps explain the heterogeneous effects observed in full multi-agent evaluations. Workflows that hinge on revision or correction are more sensitive to reviser competence, whereas workflows centered on planning or aggregation are less discriminative with respect to model choice. Overall, the analysis indicates that different agent roles place uneven demands on model capabilities, and that role-aware evaluation is necessary for interpreting multi-agent performance beyond aggregate accuracy.

Table 7. Role-specific performance comparison under controlled role-isolation settings. Reviser is evaluated on HumanEval, Aggregator on ARC-Challenge, and Planner on GSM-Hard.

Model	Reviser	Aggregator	Planner
Pangu-7B	92.07	91.38	68.69
Qwen2.5-7B	79.27	92.15	69.07
DeepSeek-7B	86.59	92.58	67.93

6.2. Inference Cost and Accuracy Trade-offs

We analyze the cost–accuracy trade-offs of different MAS workflows under the `no_think` strategy, using ARC-Challenge as a representative benchmark, with total token consumption serving as a proxy for inference cost (Sec. 3.3). While we focus on ARC-Challenge here, analogous analyses on additional benchmarks are reported in Appendix C and exhibit consistent qualitative trends.

Fig. 3 summarizes the overall cost-effectiveness frontier: *Reflection* achieves the highest success rate while maintaining a low mean token cost, indicating that lightweight post-hoc correction can yield substantial quality gains without triggering large context growth. *Interactive Debate* attains a comparable success rate but at a much higher average cost, suggesting diminishing returns when additional interaction rounds primarily add redundancy rather than decisive evidence. In contrast, *Adversarial Debate* has the highest mean token cost while achieving only a mid-tier success rate, substantially trailing *Reflection* and *Interactive Debate*. Its extremely wide token range further suggests highly variable compute demand across instances, weakening its practical cost–reliability profile. *Plan-and-Execute* operates at a similarly low token budget to *Reflection*, but yields the lowest success rate among all methods on ARC-Challenge, indicating that the added structure does not translate into competitive accuracy in this setting.

Fig. 4 reveals that token cost is only weakly explained by input length. For all methods, token usage exhibits substantial dispersion at similar query lengths, implying that the dominant driver of cost is *strategy-induced interaction dynamics* (e.g., number of turns, verbosity cascades, and transcript accumulation) rather than the query itself. Notably, debate-style methods exhibit a clear heavy-tail regime: a subset of instances triggers extremely long generations (up to $\sim 7 \times 10^4$ tokens), reflecting a practical risk of cost blow-up under adversarial or multi-party exchanges. By comparison, *Reflection* shows a much tighter band with limited outliers, indicating better cost controllability.

Finally, Fig. 5 analyzes inference cost at the instance level. The token distribution is clearly bimodal, with a low-cost mode (roughly 2–4K tokens) and a high-cost mode (around 6–10K tokens), revealing substantial heterogeneity across problem instances. Crucially, failed cases are

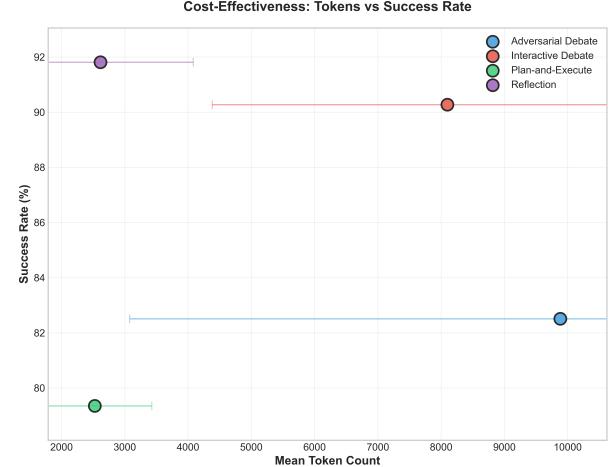


Figure 3. Mean token cost and success rate across MAS workflows on ARC-Challenge.

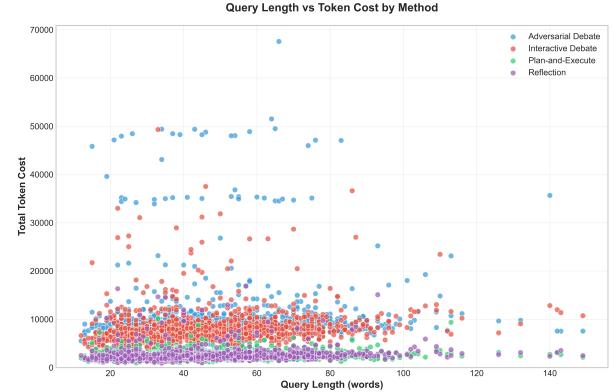


Figure 4. Query length versus total token cost for different MAS workflows on ARC-Challenge.

heavily concentrated in the high-cost regime. This indicates that elevated token consumption is not a signal of additional reasoning paying off, but rather a manifestation of the model struggling on inherently difficult instances—where more computation is expended without resolving the underlying uncertainty.

7. Conclusion

This work presents a comprehensive investigation into the landscape of reasoning paradigms for LLMs, spanning from direct single-model generation and CoT augmentation to representative MAS. Our analysis reveals a critical trade-off: increased structural complexity does not guarantee improved reasoning. By evaluating these paradigms within a unified framework—integrating closed-form benchmarks with the novel evaluation axis introduced by our MIMeBench—we clarify the circumstances under

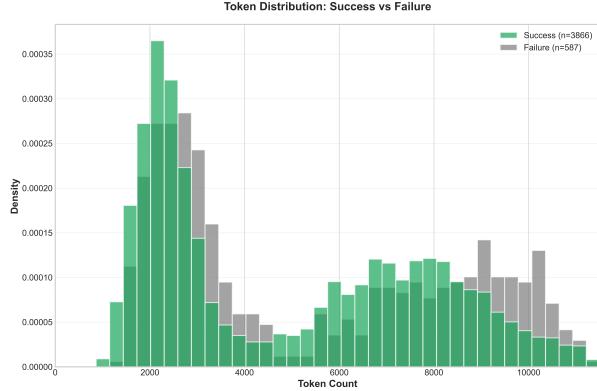


Figure 5. Token cost distributions for successful and failed instances on ARC-Challenge.

which structural complexity provides meaningful improvements, as opposed to cases where it yields limited or unstable gains. Ultimately, our findings provide a principled guide for the design and deployment of LLM-based reasoning systems, clarifying the intricate relationship between paradigm choice, performance reliability, and operational efficiency.

However, our study has several limitations. The analysis is mainly conducted on Pangu-7B model and a limited set of representative workflows, and the extent to which these findings generalize to other architectures or agent designs remains an open question. In addition, inference efficiency is primarily measured by token usage, which does not fully capture system-level latency or hardware constraints. Future work will extend this investigation to a broader range of models, and incorporate more comprehensive efficiency metrics to strengthen the empirical grounding of these findings.

References

- [1] Hanting Chen, Yasheng Wang, Kai Han, Dong Li, Lin Li, Zhenni Bi, Jinpeng Li, Haoyu Wang, Fei Mi, Mingjian Zhu, et al. Pangu embedded: An efficient dual-system llm reasoner with metacognition. *arXiv preprint arXiv:2505.22375*, 2025. [2](#)
- [2] Justin Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, 2024. [3](#)
- [3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code, 2021. [2](#)
- [4] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *ICLR*, 2024. [1, 3](#)
- [5] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, et al. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018. [2](#)
- [6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. [2](#)
- [7] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. [7](#)
- [8] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023. [1, 3](#)
- [9] Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, 2024. [2](#)
- [10] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, et al. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*, 2022. [2](#)
- [11] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2023. [3](#)
- [12] Pranjal Kumar. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260, 2024. [1](#)
- [13] Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024. [3](#)
- [14] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904, 2024. [3](#)
- [15] Shalev Lifshitz, Sheila A McIlraith, and Yilun Du. Multi-agent verification: Scaling test-time compute with multiple verifiers. *arXiv preprint arXiv:2502.20379*, 2025. [2, 3](#)
- [16] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. [2](#)
- [17] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, 2017. [2](#)
- [18] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [2](#)

- [19] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023. [2](#)
- [20] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Halilinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023. [1](#), [2](#)
- [21] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024. [1](#)
- [22] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, et al. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. [2](#)
- [23] Matthew Renze and Erhan Guven. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*, 2024. [2](#)
- [24] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023. [2](#)
- [25] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. [2](#)
- [26] Qwen Team. Qwen3 technical report, 2025. [7](#)
- [27] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025. [3](#)
- [28] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*. [2](#)
- [29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. [1](#)
- [30] Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. Answering questions by meta-reasoning over multiple chains of thought. In *The 2023 Conference on Empirical Methods in Natural Language Processing*. [2](#)
- [31] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023. [1](#)
- [32] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023. [2](#)

Appendix

A. Role-Isolation Evaluation Protocol

All experiments are conducted using open-weight models, with Pangu-7B, Qwen2.5-7B, and DeepSeek-7B evaluated under identical role-isolation settings. For each target role, the evaluated model is substituted into that role while all other components of the multi-agent workflow are held fixed, enabling controlled comparison across models.

For MAS that involve intermediate reasoning artifacts, including Reflection and Interactive Debate, we adopt a fixed-context evaluation protocol. Specifically, all intermediate outputs(initial solution and debate messages) corresponding to non-target roles are generated once by a reference model (Qwen2.5-7B) and cached. The evaluated model is then applied only to the target role and operates solely on these fixed artifacts. This design ensures that different models receive identical role-specific inputs, isolating role competence from variability introduced by multi-agent interactions. An illustration of this role-isolation setup is shown in Fig. 13.

For the Plan-and-Execute workflow, fixed intermediate artifacts are not used, as the execution stage depends directly on the planner’s output. Instead, to ensure fairness and reduce stochastic effects, the Executor is run with decoding temperature set to zero when evaluating planner-related behavior, so that execution differences are attributable solely to the planner’s output.

Across all role-isolation experiments, evaluation metrics and judging procedures are kept consistent with the main experiments. For each role, performance is measured on a representative benchmark aligned with the role’s functional responsibility (HumanEval for Reviser, ARC-Challenge for Aggregator, and GSM-Hard for Planner), allowing focused and interpretable role-level comparison.

B. Prompt Template

Fig. 6–12 show prompt templates of our study.

C. Cost and Accuracy Analysis

As shown in Fig. 14–16 for GSM-Hard and Fig. 17–19 for HumanEval, both benchmarks exhibit trends that are qualitatively consistent with those observed on ARC-Challenge.

In addition, Fig. 20 serves as a supplementary cost analysis under the auto-think strategy, extending the no-think results presented earlier. Consistent with previous observations, failed instances consume substantially more tokens than successful ones across all workflows, indicating that higher inference cost remains associated with reasoning instability rather than improved outcomes, even when internal deliberation is enabled. This confirms that the

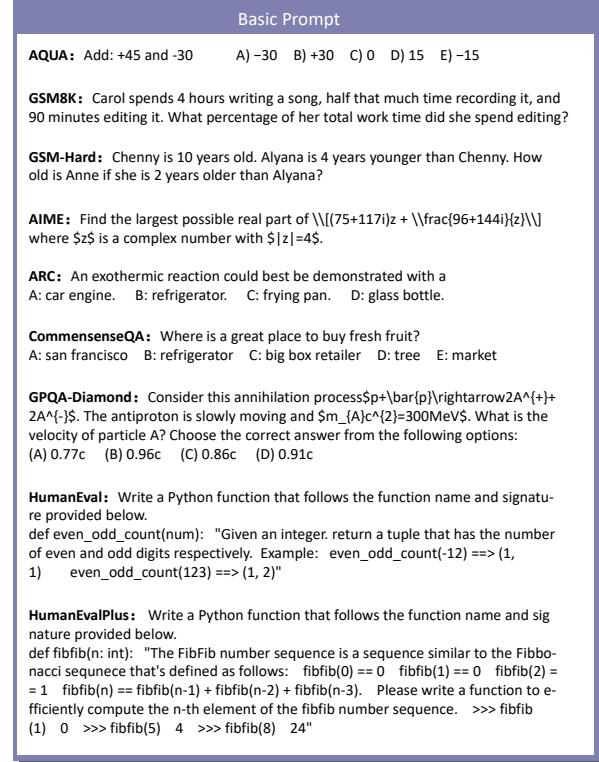


Figure 6. Basic prompt template (single-model inference).

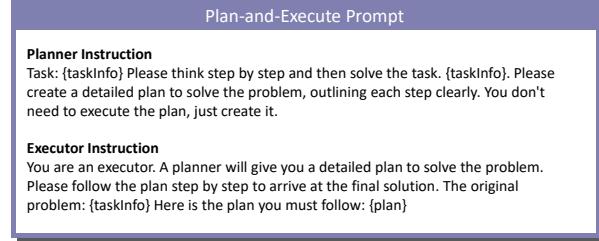


Figure 7. Plan-and-Execute prompt templates (Planner & Executor).

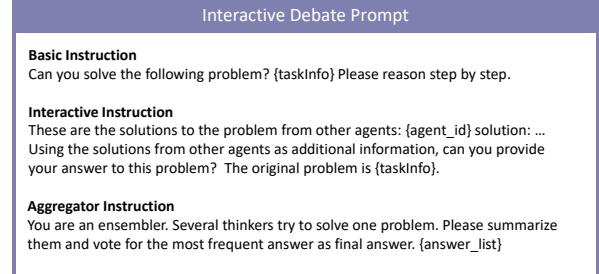


Figure 8. Interactive Debate prompt templates (Debaters & Aggregator).

Reflection Prompt

Reasoner Instruction
Task: {taskInfo} Please think step by step and then solve the task.

Reviser-Feedback Instruction
You are a refiner. Here is an initial answer to a problem:--- Initial Answer - {initial_answer}--- End of Initial Answer --- Please check the answer carefully to see if there are any errors in logic, common sense, reasoning, etc. If so, please point out the errors and give suggestions for improvement. The original problem was: {taskInfo}

Reviser-Refine Instruction
Based on your own feedback and critique, please refine the initial answer. Provide the improved and final solution.

Figure 9. Reflection prompt templates (Reasoner & Reviser).

Adversarial Debate Prompt

Debater-System Prompt
You are a debater. Hello and welcome to the debate. It's not necessary to fully agree with each other's perspectives, as our objective is to find the correct answer. The debate topic is stated as follows: {taskInfo}

Debater-Instruction
###opponent_ans###\nDo you agree with my perspective? Please provide your reasons and answer.

Base_Answer-Instruction
Task: {taskInfo}Please think step by step and then solve the task.

Aff-Instruction
You think the correct answer is: ###base_answer### Restate the answer and provide your reasons.

Neg-Instruction
###aff_ans### You disagree with my answer. Provide your answer and reasons.

Judge-System Prompt
You are a moderator. There will be two debaters involved in a debate. They will present their answers and discuss their perspectives on the following topic: {taskInfo} At the end of each round, you will evaluate answers and decide which is correct.

Judge-Instruction
Now one round of debate for both sides has ended. Affirmative side arguing: ###aff_ans### Negative side arguing: ###neg_ans### You, as the moderator, will evaluate both sides' answers and determine if there is a clear preference for an answer candidate. Finally, please summarize your reasons for supporting affirmative/negative side and give the final answer that you think is correct.

Figure 10. Adversarial Debate prompt templates (Affirmative, Negative, Judge).

cost-accuracy patterns identified under no-think persist under auto-think, reinforcing the robustness of our conclusions.

D. Case Study

Figure 21 provides a concrete example of the Interactive Debate process on ARC-Challenge under the `auto_think` strategy, illustrating the results discussed in Table 5.

Automated Judge(Qwen3-32B) Prompt

Extract Code Instruction
Task: {taskInfo} Please think step by step and then solve the task. {taskInfo}. Please create a detailed plan to solve the problem, outlining each step clearly. You don't need to execute the plan, just create it.
You are given a **Solution**. The **Solution** gives a python function to solve a code task. Extract the final code function from the **Solution**.
Solution: {solution}

Please follow the following rules:

- Only output the code function that exists in the **Solution**, without any additional explanation or content.
- Do not modify any part of the code function.
- Remove parts like 'example use' or 'test cases'.
- If the **Solution** does not contain a code function, respond with: "The reply doesn't contain a code function."
- If the **Solution** contains a code function. You **must** output python code function wrapped by ```python\n ... \n```, like ```python\ndef fibfib(n: int):\n if n == 0:\n return 0\n if n == 1:\n return 1\n if n == 2:\n return 1\n return fibfib(n - 1) + fibfib(n - 2) + fibfib(n - 3)\n```.

Judge Instruction
Please judge whether the given answer is correct or incorrect based on the ground truth.\nGiven Answer: {extract_answer}\nGround truth: {gt}\nThe given answer do not need to match ground truth perfectly, as long as they are consistent, you can think the given answer is correct. For example, given answer: two is correct based on the ground truth: 2.0, given answer: 3.36 is correct based on the ground truth: \frac{37}{11} and so on. If the given answer is correct, output \boxed{{1}} at the end of your response, otherwise output \boxed{{0}} at the end of your response.

Figure 11. Automated judge prompt templates (Qwen3-32B) for evaluation.

Passage: <text>

Question: <text>

Constraints: Each option must be 10–20 Chinese characters; options should be complete sentences; do not introduce core concepts that do not appear in the passage; avoid overly absolute wording, etc.

Output:
A. ... / B. ... / C. ... / D. ...

Figure 12. Illustrative prompt template for models evaluated on MIMeBench (fields only; no real content). Input includes the passage, question, optional constraints (e.g., length or format), and the required output format; the model should output four structured options (A–D) while satisfying length and style constraints.

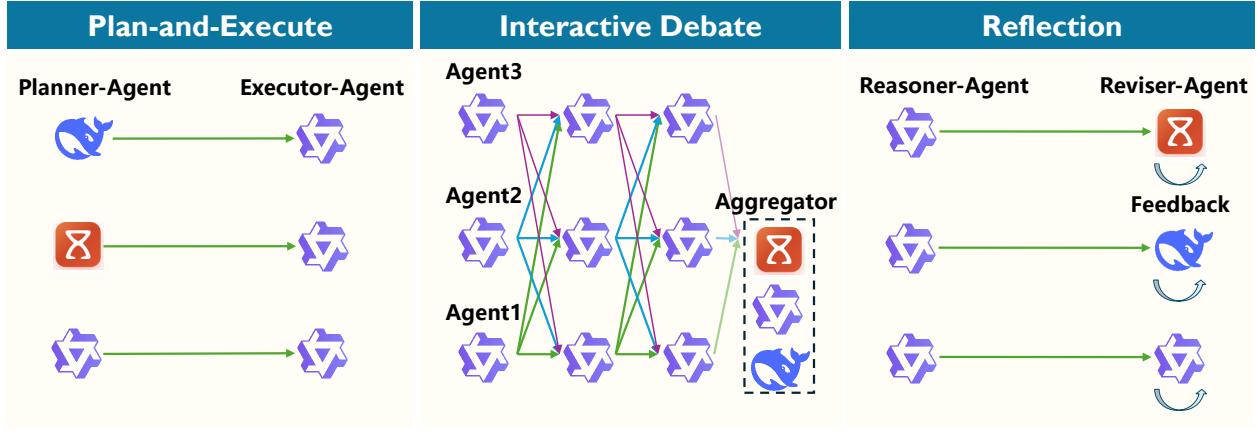


Figure 13. Role-Isolated Evaluation Workflow for Multi-Agent Systems.

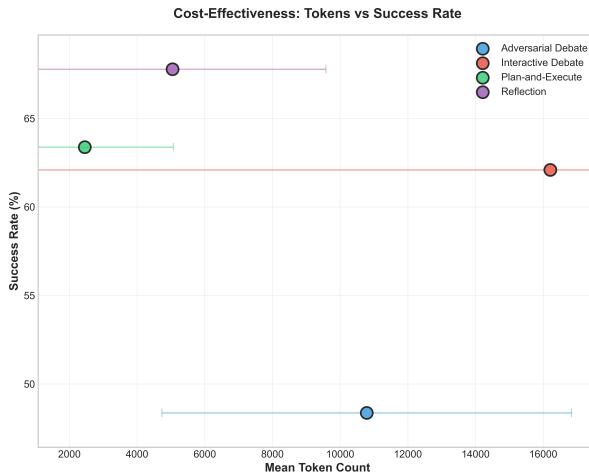


Figure 14. Mean token cost and success rate across MAS workflows on GSM-Hard.

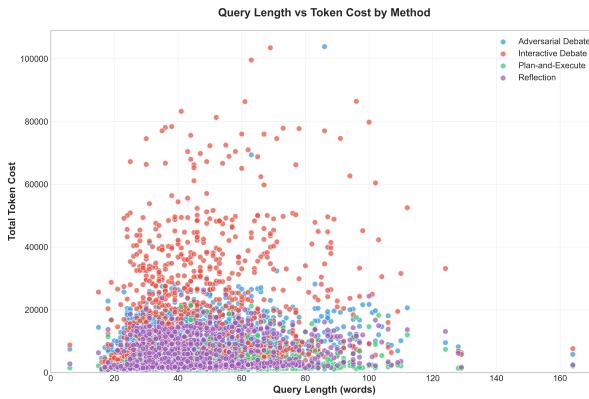


Figure 15. Query length versus total token cost for different MAS workflows on GSM-Hard.

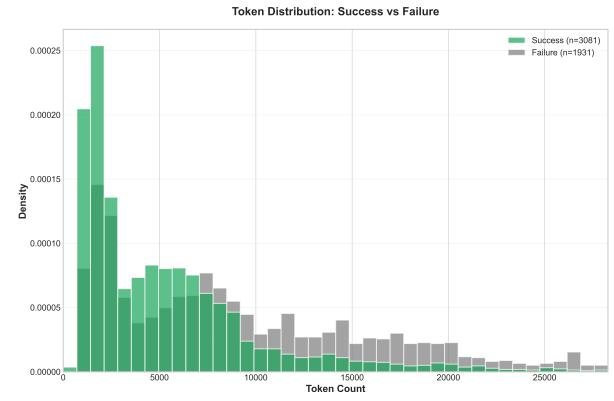


Figure 16. Token cost distributions for successful and failed instances on GSM-Hard.

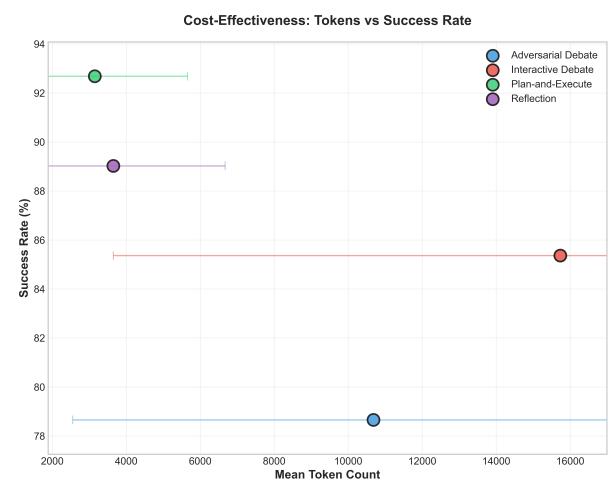


Figure 17. Mean token cost and success rate across MAS workflows on HumanEval.

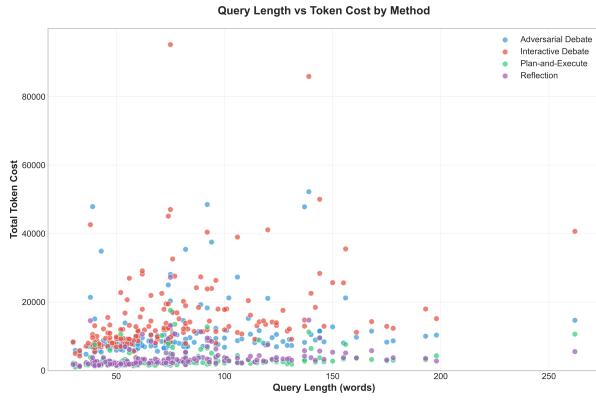


Figure 18. Query length versus total token cost for different MAS workflows on HumanEval.

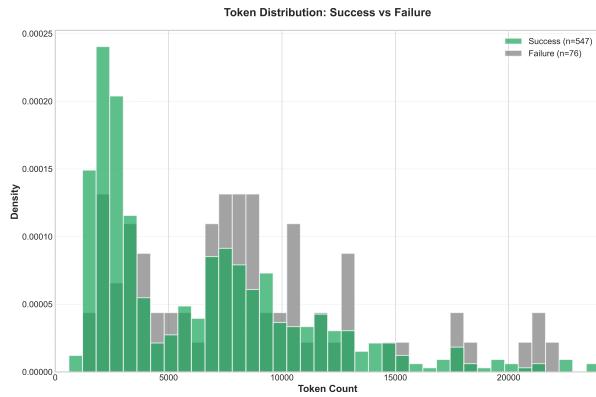


Figure 19. Token cost distributions for successful and failed instances on HumanEval.

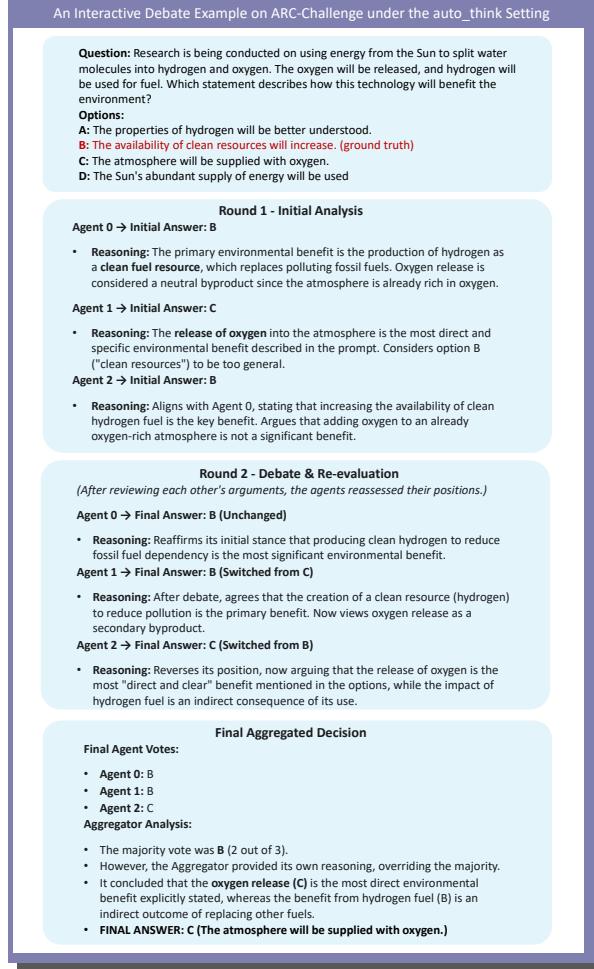


Figure 21. An example of interactive debate on the ARC-Challenge task under the `auto_think` strategy. Three agents first generate independent answers by a CoT procedure, followed by a debate and re-evaluation phase. While the majority of agents favor the clean-resource interpretation, the ensemble ultimately selects an alternative option based on explicit semantic alignment. This case illustrates that, when a strong CoT procedure is already present, additional multi-agent interactions may lead to inconsistent outcomes and diminished returns.

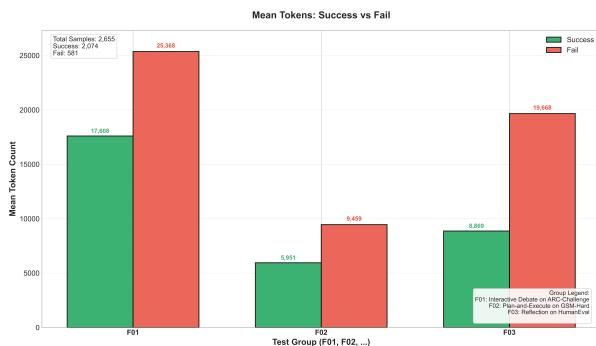


Figure 20. Mean token cost for successful and failed instances under `auto_think` strategy.