

TEAM OF THOUGHTS: EFFICIENT TEST-TIME SCALING OF AGENTIC SYSTEMS THROUGH ORCHESTRATED TOOL CALLING

Anonymous authors

Paper under double-blind review

ABSTRACT

Existing Multi-Agent Systems (MAS) typically rely on static, homogeneous model configurations, limiting their ability to exploit the distinct strengths of differently post-trained models. To address this, we introduce **Team-of-Thoughts**, a novel MAS architecture that leverages the complementary capabilities of heterogeneous agents via an orchestrator-tool paradigm. Our framework introduces two key mechanisms to optimize performance: (1) an orchestrator calibration scheme that identifies models with superior coordination capabilities, and (2) a self-assessment protocol where tool agents profile their own domain expertise to account for variations in post-training skills. During inference, the orchestrator dynamically activates the most suitable tool agents based on these proficiency profiles. Experiments across five reasoning and code generation benchmarks demonstrate that Team-of-Thoughts achieves a superior accuracy-cost trade-off. Notably, on AIME24, our method reaches 96.67% accuracy, significantly surpassing homogeneous role-play baselines (80%) while reducing computational costs by two orders of magnitude.

1 INTRODUCTION

Test-time scaling (TTS) has emerged as a critical paradigm for enhancing the capabilities of large language models (LLMs) beyond their training-time performance (Snell et al., 2024; Wu et al., 2025; Brown et al., 2024). By investing additional computation during inference, TTS methods such as process reward model (PRM) scoring, beam search, and tree-based exploration (Wei et al., 2023; Yao et al., 2023; Besta et al., 2024) enable models to achieve superior performance on complex reasoning tasks. This paradigm shift recognizes that thoughtful allocation of inference-time compute can unlock latent capabilities within pre-trained models, making TTS a fundamental technique for deploying LLMs in high-stakes applications requiring robust reasoning and problem-solving abilities.

While existing TTS approaches have demonstrated impressive gains, they typically operate within the confines of a single model or rely on static multi-agent workflows with fixed role assignments of the same model (Park et al., 2023; Qian et al., 2024; Hong et al., 2024; Li et al., 2023; Wu et al., 2023). This limitation prevents systems from exploiting the complementary strengths of diverse LLMs, each of which may excel in different domains due to distinct post-training procedures and dataset composition. Moreover, conventional TTS methods often scale inefficiently, generating excessive tokens without strategic allocation of computational resources. Recent multi-agent systems (Zhang et al., 2024; Chen et al., 2024; Yang et al., 2025b; Li et al., 2025; Kim et al., 2026), while introducing parallelism, still lack the flexibility to dynamically adapt agent selection and orchestration based on task characteristics and model-specific expertise.

We address these limitations through Team-of-Thoughts, a novel Multi-Agent System (MAS) that achieves efficient test-time scaling via orchestrated tool calling. Unlike conventional approaches that treat models as monolithic reasoners or assign them to rigid roles, our framework reconceptualizes diverse LLMs as specialized tools that can be dynamically invoked and coordinated. The key insight is that by leveraging the native tool-calling capabilities of modern LLMs, we can build a hierarchical architecture where an orchestrator agent strategically activates and allocates computational budget to a team of specialized tool agents, as demonstrated in Figure 1.

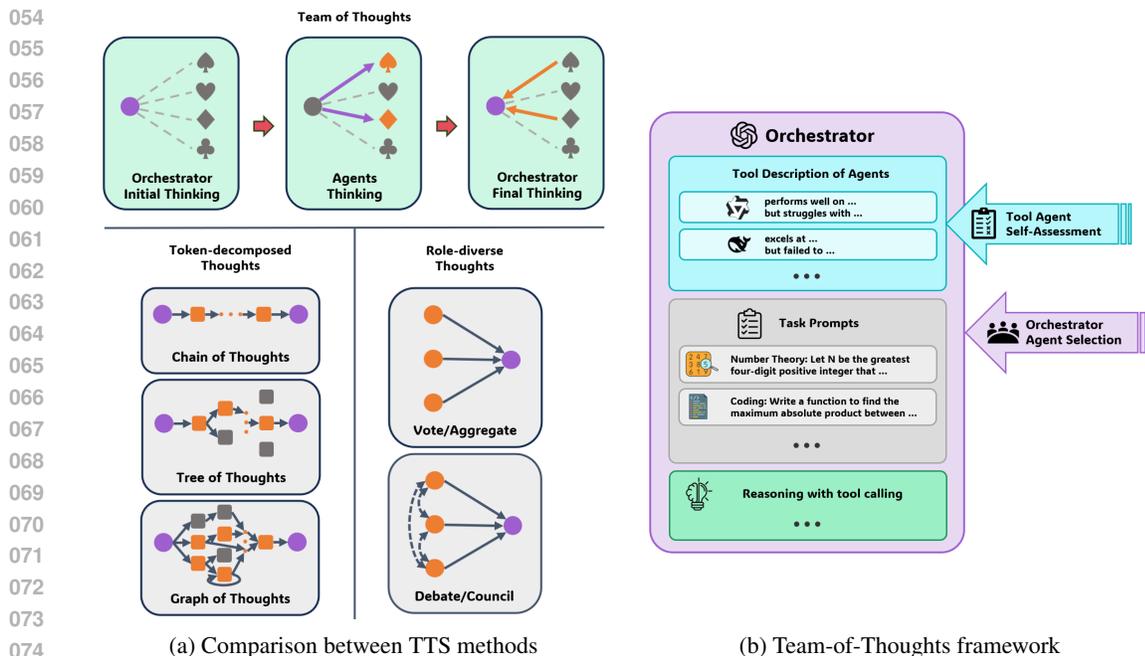


Figure 1: **Overview of Team-of-Thoughts.** (a) While standard reasoning methods rely on a single model (Token-decomposed Thoughts) or homogeneous multi-agent groups (Role-diverse Thoughts), Team-of-Thoughts incorporates heterogeneous models to ensure broad coverage of the solution space. (b) Our framework integrates an orchestrator for tool-agent management, utilizing an initialization pipeline that includes orchestrator calibration and agent self-profiling. At inference time, the orchestrator identifies the optimal tools for the input query and synthesizes their reasoning trajectories into a high-confidence final response.

This design naturally enables efficient parallelism, multiple agents can reason simultaneously, while maintaining token efficiency through selective activation based on task requirements and agent expertise. Our framework transforms test-time scaling from a sequential, token-heavy process into a coordinated team effort where the right specialists are consulted at the right time.

Our key contributions are as follows:

- We propose the Team-of-Thoughts MAS, that enables multiple specialized LLMs to collaborate via tool-calling interfaces without redundant reasoning traces.
- We propose an orchestration calibration scheme that identifies the optimal orchestration agent for coordinating tool agents, revealing significant variation in orchestration capabilities across model families. We also develop a self-assessment mechanism that captures agent specialization, allowing tool agents to self-estimate their proficiency across task categories and enabling informed decisions about agent activation and budget allocation.
- We conduct extensive experiments across multiple model families and demonstrate that Team-of-Thoughts MAS achieves a superior trade-off between accuracy and computational cost on both reasoning and code generation tasks. For example, our method reaches 96.67% accuracy on AIME24, outperforming AgentVerse’s 80%, while incurring a computational cost that is two orders of magnitude lower.

2 BACKGROUND

2.1 A PROBABILISTIC VIEW ON TTS

A task-solving problem can be formulated as a tuple (D, X) , where D is the input question and X is the target answer. The objective of a language model is to optimize its probabilistic distribution

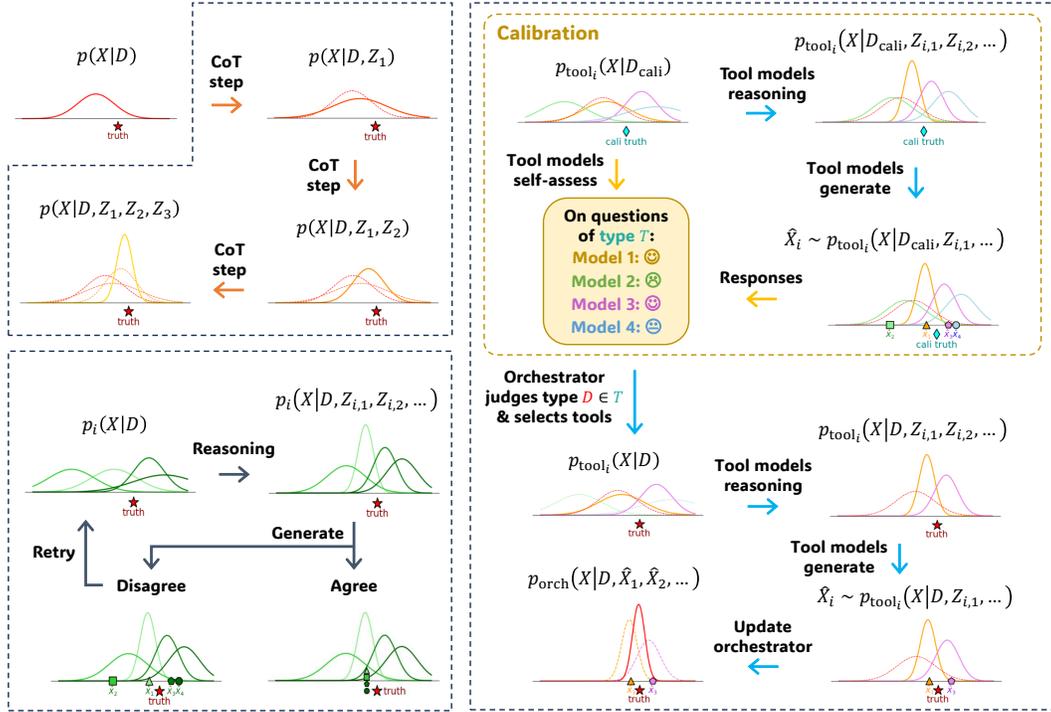


Figure 2: **Schematic comparison of language modeling methods.** (Top left) Standard Inference: A single model predicts target X directly from input D . (Top middle) Agentic Reasoning: Methods like CoT generate intermediate steps to refine the prediction distribution. (Bottom left) Consensus-based MAS: Multiple agents reason iteratively until consensus is reached. (Right) Team-of-Thoughts MAS: An orchestrator leverages heterogeneous tool agents. During calibration, agents self-assess their proficiency on question types T . During inference, the orchestrator selectively invokes agents based on these assessments, aligning the prediction with the target while maintaining token efficiency.

$p_{\theta}(\cdot|D)$ to maximize the likelihood of generating the correct prediction:

$$\hat{X} \sim p_{\theta}(\cdot|D)$$

as illustrated in the top-left panel of Figure 2.

To enhance performance beyond standard inference, agent models leverage reasoning frameworks, such as Chain-of-Thought (CoT) (Wei et al., 2023), Tree-of-Thoughts (ToT) (Yao et al., 2023), and Graph-of-Thoughts (GoT) (Besta et al., 2024), to generate intermediate reasoning steps. Each step, as well as the final prediction, is conditioned on the preceding context:

$$\begin{aligned} Z_1 &\sim p_{\theta}(\cdot|D), \\ Z_2 &\sim p_{\theta}(\cdot|D, Z_1), \\ &\vdots \\ \hat{X} &\sim p_{\theta}(\cdot|D, Z_1, Z_2, \dots) \end{aligned}$$

where Z_i denotes the i th intermediate thinking step. These reasoning steps explore the generation space, ideally shifting the prediction distribution toward the target \hat{X} , as shown in the top-middle panel of Figure 2.

However, a single agent’s ability to transform the prediction distribution is strictly bounded by its parameterization θ . Facing increasingly complex real-world tasks, a single model may fail to reach the solution space. While test-time scaling approaches (Snell et al., 2024; Wu et al., 2025; Brown et al., 2024) invest additional compute to improve single-model performance, they remain fundamentally limited by the model’s fixed parameters. Training larger models remains prohibitively expensive, motivating alternative approaches.

2.2 EXTENDING TO MAS

The limitation of single-model TTS motivates Multi-Agent Systems (MAS), which employ multiple expert agents to create a more robust system. Instead of relying on a single model parameterization θ , MAS leverage an ensemble of models $\{\theta_1, \theta_2, \dots, \theta_n\}$, each starting with a distinct prior distribution over the solution space.

Formally, each agent i begins with its own prediction distribution $p_i(X|D) = p_{\theta_i}(X|D)$, representing different model priors as shown in the bottom-left panel of Figure 2. Through iterative reasoning, each agent refines its distribution:

$$p_i(X|D, Z_{i,1}, Z_{i,2}, \dots)$$

where $Z_{i,j}$ denotes the j -th reasoning step of agent i .

In a consensus-based MAS (Chen et al., 2024), agents sample predictions $\{\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n\}$ from their respective distributions and communicate to reach agreement. When agents disagree, they exchange reasoning and retry; when they agree, the system outputs the consensus. Ideally, this iterative process concentrates probability mass around the correct answer through the exchange of diverse perspectives, as shown in Figure 2.

Many approaches construct agents by prompting a single underlying model with different personas or instructions (Chen et al., 2024; Yang et al., 2025b; Li et al., 2025; Zhang et al., 2024; Wu et al., 2023; Li et al., 2023; Hong et al., 2024). Mathematically, this means all agents share the same fundamental parameterization: $\theta_1 = \theta_2 = \dots = \theta_n = \theta$. Consequently, their priors $p_i(X|D)$ remain largely identical, and the system fails to effectively diversify the prediction distributions. As shown in the bottom-left panel of Figure 2, one hypothesis we hold is that true diversity requires heterogeneous model priors – different models with distinct parameterizations that cover complementary regions of the solution space.

Moreover, consensus-based approaches suffer from inefficiencies: they require multiple rounds of generation, retry mechanisms, and full reasoning traces from all agents regardless of their relevance to the specific task. This motivates our Team-of-Thoughts framework, which leverages heterogeneous model priors and introduces an orchestrator agent to strategically coordinate tool agents, as shown in the right panel of Figure 2.

3 TEAM OF THOUGHTS: AN EFFICIENT HETEROGENEOUS MAS APPROACH

Inspired by the formulation in Section 2, our **Team-of-Thoughts** framework composes a MAS using heterogeneous agent models to achieve broad capability coverage. As illustrated in Figure 2 (right), the reasoning process of the tool agents begins with distinct prediction distributions $\{p_{\theta_1}(X|D), p_{\theta_2}(X|D), \dots, p_{\theta_n}(X|D)\}$, ensuring extensive coverage of the solution space.

The Team-of-Thoughts framework constitutes an orchestrator agent and a team of tool agents. The orchestrator’s role is threefold: (1) selecting tool agents suitable for the incoming question D , (2) evaluating the responses and analyzing the reasoning of the tool agents, and (3) conducting its own reasoning and acting as an aggregator to generate the final answer. Each tool agent i , when invoked, produces a reasoning trajectory $Z^{(i)} = \{Z_{i,1}, Z_{i,2}, \dots\}$ and generates a prediction $\hat{X}_i \sim p_{\theta_i}(X|D, Z^{(i)})$. The orchestrator then updates its prediction distribution by aggregating information from the selected tool agents:

$$\hat{X}_{\text{orch}} \sim p_{\text{orch}}(X|D, \hat{X}_1, \hat{X}_2, \dots, \hat{X}_k) \quad (1)$$

where $k \leq n$ denotes the number of tool agents invoked. Unlike consensus-based MAS that requires agreement among all agents, the orchestrator strategically weights and filters information based on agent expertise and task characteristics, effectively concentrating probability mass on the target X while maintaining token efficiency.

3.1 ORCHESTRATION CALIBRATION

Not all models are equally capable of orchestration. The orchestrator must understand tool capabilities, coordinate multiple agents, synthesize diverse reasoning trajectories, and resolve conflicts—skills

that vary significantly across model families. To identify the optimal orchestrator, we propose an **Orchestration Calibration** procedure. We evaluate candidate orchestrator models based on their ability to aggregate tool-agent responses on a calibration set for a given task category c under a fixed cost budget. For each candidate orchestrator model θ_{cand} and task category c , we measure orchestration performance on the category-specific calibration dataset:

$$\text{Score}(\theta_{\text{cand}}, c) = \frac{1}{|D_{\text{val}}^{(c)}|} \sum_{D \in D_{\text{val}}^{(c)}} \mathbb{I}[\hat{X}_{\text{cand}}(D) = X(D)]$$

where $D_{\text{val}}^{(c)}$ denotes the calibration dataset for category c , $\hat{X}_{\text{cand}}(D)$ is the final aggregated prediction in Equation (1), $X(D)$ is the ground-truth answer, and $\mathbb{I}[\cdot]$ is the indicator function. We select the candidate model with the highest calibration score as the orchestrator for category c .

We show in Section 4.1 that this calibration process reveals that orchestration capability does not simply correlate with model size or general benchmark performance. Some models excel at reasoning integration and strategic decision-making, while others perform better as specialized tool agents. This finding motivates treating the orchestrator selection as a distinct optimization problem rather than defaulting to the largest or most capable model.

3.2 AGENT SPECIALIZATION VIA SELF-ASSESSMENT

Different model families exhibit varying expertise across task categories due to their distinct post-training procedures. To leverage this specialization, we introduce a **self-assessment mechanism** that allows tool agents to estimate their proficiency on different task types.

For each tool agent i and task category c , we compute a proficiency score on a validation set:

$$s_i^{(c)} = \frac{1}{|D_{\text{val}}^{(c)}|} \sum_{D \in D_{\text{val}}^{(c)}} \mathbb{I}[\hat{X}_i(D) = X(D)]$$

where $D_{\text{val}}^{(c)}$ contains validation examples from category c . These proficiency scores form a specialization profile for each agent:

$$\mathbf{s}_i = [s_i^{(1)}, s_i^{(2)}, \dots, s_i^{(C)}]$$

where C is the number of task categories. At inference time, given a query D classified into category c , the orchestrator can selectively activate tool agents based on their proficiency scores $s_i^{(c)}$. This enables strategic budget allocation: highly proficient agents for the task category receive priority, while less relevant agents may be skipped entirely. This approach contrasts with static MAS that invoke all agents regardless of task fit.

3.3 EFFICIENT PARALLELISM AND TOKEN SCALING

Our framework achieves efficient test-time scaling through two key mechanisms:

Agent Parallelism. Unlike sequential MAS like Zhang et al. (2024) and Li et al. (2025), tool agents in Team-of-Thoughts can reason simultaneously. When the orchestrator invokes multiple tool agents in a round, their generations can be parallelized, significantly reducing latency compared to sequential processing.

Strategic Token Allocation. The orchestrator does not require full reasoning traces from all agents. It only dynamically picks a subset of agents to join the task. Taking self-assessment profiles and task characteristics, the orchestrator can:

1. Skip tool agents with low proficiency for the task category;
2. Request shorter responses from less critical agents;
3. Allocate more token budget to highly specialized agents.

Formally, let N_i denote the number of tokens generated by tool agent i . The total token budget is:

$$N_{\text{total}} = N_{\text{orch}} + \sum_{i \in S} N_i$$

Table 1: Main results of Team of Thoughts (ToT) on five general benchmark tasks. LiveCodeBench is using v6 (2025/01/01 - 2025/05/01) data. We report task accuracy (%,"Acc.") and total execution cost per task ("Cost"). We compare ToT with single-model baselines and multi-agent methods. "Improve" compares ToT with the best baseline accuracy.

Method	AIME24		AIME25		HumanEval+		MBPP+		LiveCodeBench v6	
	Acc (↑)	Cost (↓)	Acc (↑)	Cost (↓)						
Claude Sonnet 4.5	76.67%	0.4206	66.67%	0.4467	95.73%	0.9057	83.33%	1.6272	51.10%	1.6116
GPT-5 Mini	86.67%	0.0352	83.33%	0.0395	92.07%	0.1428	81.48%	0.2442	65.93%	0.1371
Gemini 3 Flash	86.67%	0.2257	73.33%	0.2931	70.12%	0.1285	69.31%	0.2792	70.33%	0.4844
DeepSeek v3.2	86.67%	0.0062	90.00%	0.0068	87.80%	0.0163	77.51%	0.0328	65.93%	0.0356
GPT-OSS 20B	60.00%	0.0011	66.67%	0.0040	87.20%	0.0074	76.46%	0.0131	50.00%	0.0092
Qwen3-v1 235B	83.33%	0.0848	86.67%	0.1186	87.80%	0.3213	80.69%	0.6524	65.93%	0.6573
Phi-4	10.00%	0.0100	20.00%	0.0050	71.95%	0.0108	63.49%	0.0203	24.18%	0.0209
AgentVerse	80.00%	2.5302	60.00%	4.1131	91.46%	7.7689	79.37%	14.1327	65.93%	14.5509
Majority Voting	93.33%	0.7836	90.00%	0.9137	–	–	–	–	–	–
Team-of-Thoughts	96.67%	0.0906	90.00%	0.0852	96.34%	0.5175	83.60%	1.7225	72.53%	1.1135

Table 2: Calibration accuracy across orchestrator model choices under different budget constraints. We report calibration accuracy (%) on AIME2024 and MBPP+ with two budget settings (USD) and their average ("Avg"). Results compare multiple language models used as the orchestrator agent. For each benchmark, we bold the best average performance across models.

Model	AIME2024			MBPP+			
	Budget Cost (\$)	0.03	0.02	Avg	0.03	0.02	Avg
Claude Sonnet 4.5	86.67%	46.67%	66.67%	66.67%	78.38%	78.38%	78.38%
GPT-5 Mini	86.67%	93.33%	90.00%	90.00%	86.49%	83.78%	85.14%
Gemini 3 Flash	86.67%	40.00%	63.34%	63.34%	83.78%	83.78%	83.78%
DeepSeek v3.2	93.33%	93.33%	93.33%	93.33%	81.08%	81.08%	81.08%
GPT-OSS 20B	80.00%	80.00%	80.00%	80.00%	75.68%	75.68%	75.68%
Qwen3-v1 235B	33.33%	20.00%	26.67%	26.67%	78.38%	78.38%	78.38%
Phi-4	33.33%	33.33%	33.33%	33.33%	72.97%	72.97%	72.97%

where S is the set of activated tool agents and N_{orch} is the orchestrator’s token usage. By strategically controlling $|S|$ and the generation lengths $\{N_i\}_{i \in S}$, Team-of-Thoughts achieves superior performance-to-token ratios compared to both single-model TTS (which cannot leverage diverse priors) and consensus-based MAS (which invoke all agents indiscriminately).

4 EVALUATION

Models and benchmarks We evaluated Team-of-Thoughts MAS across a diverse suite of seven model families, comprising three closed-source models: Claude-Sonnet-4.5 (PBC, 2025), GPT-5-mini (Singh et al., 2025), and Gemini-3-Flash-Preview (Pichai et al., 2025) and four open-source models: DeepSeek-V3.2-Exp (DeepSeek-AI et al., 2025), GPT-OSS-20B (OpenAI et al., 2025), Qwen3-VL-235B-A22B-Thinking (Yang et al., 2025a), and Phi-4 (Abdin et al., 2024). Our assessment spanned two domains: mathematical reasoning (AIME2024 (of America, 2024), AIME2025 (of America, 2025)) and code generation (Humaneval+ (Chen et al., 2021), MBPP+ (Austin et al., 2021), LiveCodeBench (Jain et al., 2024)). Unless stated otherwise, we set a standardized context window for each tool-agent: 20,000 tokens for AIME tasks and 4,096 tokens for coding tasks. For the Team-of-Thoughts MAS, we used a 16,384 token context window across all tasks to ensure sufficient capacity for processing tool descriptions and making informed selection and reasoning decisions.

4.1 ORCHESTRATION AGENT SELECTION

To determine the optimal orchestrator for our framework, we first evaluated the aggregation efficacy of various candidate models. We conducted these experiments on AIME2024 and MBPP+ under fixed monetary constraints. Given the variance in pricing across providers, we normalized the budget by converting the fixed cost into model-specific token generation limits, ensuring a fair, cost-controlled

comparison. As detailed in Table 2, DeepSeek v3.2 demonstrated superior aggregation performance on the AIME2024 mathematical benchmark, whereas GPT-5 Mini excelled on the MBPP+ code generation task. Consequently, we adopted DeepSeek v3.2 as the orchestrator for mathematical reasoning and GPT-5 Mini for code generation in all subsequent analyses.

4.2 TOOL-AGENT PROFILING AND DYNAMIC SELECTION STRATEGIES

We leverage language models to generate detailed profiles of each tool agent’s strengths and weaknesses. We evaluate three distinct selection policies:

- **Random Allocation (Baseline):** Tool models are sampled uniformly at random without prior profiling.
- **Orchestrator-Based Assessment:** The orchestrator exclusively evaluates tool agents on a calibration subset to map their competencies and failure modes. We utilize DeepSeek v3.2 for math tasks and GPT-5 Mini for code generation in this role.
- **Tool Self-Assessment:** Each tool agent audits its own proficiency. Supplied with the task question, its own reasoning traces, and the ground truth, the agent identifies specific required skills and grades its performance.

These profiles enable the orchestrator to dynamically revise invocation probabilities. Crucially, the *self-assessment* policy decouples the generation of tool descriptions from the orchestrator’s specific biases. Detailed assessment prompts are provided in Appendix C.

Table 3: Comparison of tool agent profiling strategies across benchmarks. We report task accuracy (%) for different tool agent selection methods, including *single-model execution*, *tool agent self-assessment*, *orchestration-based assessment*, and *no-assessment* (random selection). Each row specifies the underlying base model used. We bold the best-performing strategy for each benchmark.

Benchmark	Single	Self-Assessment	Orch. Assessment	Random Allocation
AIME2024 (DeepSeek v3.2)	86.67%	93.33%	90.00%	90.00%
MBPP+ (GPT-5 Mini)	81.48%	83.33%	83.33%	82.27%

Table 3 details the comparative performance of these strategies with up to two active tool agents. Results indicate that both profiling-based methods significantly surpass the single-model baselines and the *random allocation* policy. Notably, *self-assessment* achieves superior accuracy on mathematical reasoning tasks while maintaining parity with *orchestrator-based assessment* on code generation. Given this dominance in reasoning domains and its ability to decouple tool profiling from orchestrator bias, we establish *self-assessment* as the default selection strategy. Additional ablation studies concerning the number of active agents are provided in Appendix B.

4.3 PERFORMANCE AND COST EFFICIENCY ANALYSIS

We expose tool agents as callable tools to the orchestrator, integrating tool descriptions derived from their self-assessment profiles. As detailed in Table 1, **Team-of-Thoughts** achieves a superior accuracy-cost trade-off across diverse reasoning and coding benchmarks. In terms of accuracy, our framework achieves state-of-the-art performance across the evaluated tasks and consistently outperforms both single-model baselines and existing multi-agent methods.

Importantly, these accuracy gains do not come at the expense of efficiency. Team-of-Thoughts maintains substantially lower execution costs than competing multi-agent systems. Compared to AgentVerse (Chen et al., 2024), our orchestrator-based aggregation achieves higher accuracy while reducing total inference cost by an order of magnitude. Compared to Majority Voting, our method matches or exceeds accuracy while requiring only a fraction of the cost, since it relies on adaptive agent selection instead of redundant parallel sampling. Overall, Team-of-Thoughts MAS effectively combines the performance gains of multi-agent collaboration with the efficiency of single-model inference, demonstrating a Pareto-efficient frontier in system design.

5 RELATED WORK

Test-Time Scaling (TTS). Recent advancements have established TTS as a critical frontier, demonstrating that increasing inference compute can yield performance gains comparable to scaling model parameters (Snell et al., 2024; Wu et al., 2025). Parallel inference-sampling approaches, such as Best-of-N (Brown et al., 2024), leverage verification to select optimal solutions from broad search spaces. Conversely, sequential approaches focus on deepening the reasoning topology. This evolution began with linear Chain-of-Thought (CoT) (Wei et al., 2023), which evokes intermediate reasoning steps, and has advanced to non-linear frameworks like Tree of Thoughts (ToT) (Yao et al., 2023) and Graph of Thoughts (GoT) (Besta et al., 2024). These methodologies enable models to perform complex cognitive operations—such as lookahead, backtracking, and information aggregation—effectively mimicking human’s “system-2” conscious thinking processes during inference.

Multi-Agent Systems (MAS). Drawing inspiration from human social dynamics, researchers have leveraged groups of agents to enhance reasoning capabilities beyond isolated models. Foundational works demonstrated that MAS could facilitate effective role-playing, relationship formation, and long-term memory in simulated environments (Park et al., 2023; Li et al., 2023). Subsequent research applied these social dynamics to complex software engineering tasks. While early frameworks like AutoGen (Wu et al., 2023) enabled agents to collaborate through flexible, conversational flows, later systems aimed to reduce hallucinations and improve reliability by encoding Standard Operating Procedures (SOPs) into the agent interactions (Qian et al., 2024; Hong et al., 2024).

More recently, the focus has shifted toward dynamic team construction and adaptive workflows. AgentVerse (Chen et al., 2024) introduces a “recruitment” mechanism where a lead agent selects specific experts based on the task instance; it employs an iterative evaluate-refine loop to ensure consensus and accuracy. Similarly, AgentNet (Yang et al., 2025b) moves beyond fixed interaction chains by utilizing a dynamic graph topology, where agents can autonomously solve tasks, transfer responsibility, or decompose problems based on real-time needs. Theoretically, the “Society of Thoughts” (Kim et al., 2026) paradigm reveals that even single-model reasoning benefits from the implicit simulation of diverse internal perspectives.

Building on the finding that divergent perspectives improve reasoning, we introduce the **Team-of-Thoughts** framework. Unlike prior works that often rely on homogeneous underlying models, our approach explicitly leverages the divergence inherent in heterogeneous agent models. By dynamically selecting suitable agents and effectively aggregating their distinct reasoning paths, our framework facilitates optimal decision-making and execution.

6 CONCLUSION

We propose the Team-of-Thoughts framework, which explicitly leverages the skill diversity of a group of heterogeneous agent models. By dynamically selecting the most suitable orchestrator and making skill-dependent tool calls that can be executed in parallel, ToT overcomes the limitations of fixed-role, sequential multi-agent systems, achieving superior token allocation for each task. Our analysis and empirical experiment across reasoning and code generation tasks demonstrates that this approach consistently achieves more efficient cost usage and higher accuracy compared to both single-model baselines and text-based multi-agent systems. This work highlights a novel paradigm for enabling heterogeneous models to collaborate effectively, paving the way for future exploration of multi-round, complex multi-agent tasks.

REFERENCES

- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL <https://arxiv.org/abs/2412.08905>.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL <https://arxiv.org/abs/2108.07732>.

- 432 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi,
433 Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph
434 of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI*
435 *Conference on Artificial Intelligence*, 38(16):17682–17690, March 2024. ISSN 2159-5399. doi: 10.
436 1609/aaai.v38i16.29720. URL <http://dx.doi.org/10.1609/aaai.v38i16.29720>.
- 437 Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and
438 Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling,
439 2024. URL <https://arxiv.org/abs/2407.21787>.
- 441 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Ka-
442 plan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen
443 Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray,
444 Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens
445 Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis,
446 Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas
447 Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher
448 Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford,
449 Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario
450 Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language
451 models trained on code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- 452 Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang
453 Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu,
454 Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and explor-
455 ing emergent behaviors. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun
456 (eds.), *International Conference on Learning Representations*, volume 2024, pp. 20094–20136,
457 2024. URL [https://proceedings.iclr.cc/paper_files/paper/2024/file/](https://proceedings.iclr.cc/paper_files/paper/2024/file/578e65cdee35d00c708d4c64bce32971-Paper-Conference.pdf)
458 [578e65cdee35d00c708d4c64bce32971-Paper-Conference.pdf](https://proceedings.iclr.cc/paper_files/paper/2024/file/578e65cdee35d00c708d4c64bce32971-Paper-Conference.pdf).
- 459 DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu,
460 Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi
461 Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li,
462 Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang,
463 Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui
464 Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li,
465 Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua
466 Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang
467 Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei
468 Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue
469 Zhang, M. S. Di, M. Y. Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou,
470 Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu
471 Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin,
472 Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang
473 Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma,
474 Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian
475 Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing
476 Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan
477 Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao
478 Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan,
479 Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao
480 Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang
481 Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan,
482 Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu,
483 Yuan Cheng, Yuan Ou, Yuanfan Xu, Yudian Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun
484 Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren,
485 Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen
Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu
Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei
Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng

- 486 Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue
487 Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui
488 Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie
489 Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang,
490 Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu
491 Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang,
492 and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL
493 <https://arxiv.org/abs/2512.02556>.
- 494 Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang,
495 Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin
496 Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative
497 framework, 2024. URL <https://arxiv.org/abs/2308.00352>.
- 498 Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando
499 Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free
500 evaluation of large language models for code, 2024. URL [https://arxiv.org/abs/2403.](https://arxiv.org/abs/2403.07974)
501 [07974](https://arxiv.org/abs/2403.07974).
- 502 Junsol Kim, Shiyang Lai, Nino Scherrer, Blaise Agüera y Arcas, and James Evans. Reasoning models
503 generate societies of thought, 2026. URL <https://arxiv.org/abs/2601.10825>.
- 504 Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel:
505 Communicative agents for "mind" exploration of large language model society. In A. Oh,
506 T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural*
507 *Information Processing Systems*, volume 36, pp. 51991–52008. Curran Associates, Inc.,
508 2023. URL [https://proceedings.neurips.cc/paper_files/paper/2023/](https://proceedings.neurips.cc/paper_files/paper/2023/file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf)
509 [file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf).
- 510 Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang,
511 Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao,
512 Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, He Zhu, Dingfeng Shi, Piaocong
513 Wang, Yeyi Guan, Xiangru Tang, Minghao Liu, Yuchen Eleanor Jiang, Jian Yang, Jiaheng Liu,
514 Ge Zhang, and Wangchunshu Zhou. Chain-of-agents: End-to-end agent foundation models
515 via multi-agent distillation and agentic rl, 2025. URL [https://arxiv.org/abs/2508.](https://arxiv.org/abs/2508.13167)
516 [13167](https://arxiv.org/abs/2508.13167).
- 517 Mathematical Association of America. AIME 2024, 2024.
- 518 Mathematical Association of America. AIME 2025, 2025.
- 519 OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin
520 Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler
521 Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen,
522 Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives,
523 Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher,
524 Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar,
525 Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman,
526 Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park
527 Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily,
528 Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath,
529 Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles,
530 Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano,
531 Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry
532 Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu
533 Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max
534 Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey,
535 Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin
536 Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney,
537 Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting
538 Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b
539 model card, 2025. URL <https://arxiv.org/abs/2508.10925>.

- 540 Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S.
541 Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the*
542 *36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New
543 York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi:
544 10.1145/3586183.3606763. URL <https://doi.org/10.1145/3586183.3606763>.
- 545 Anthropic PBC. Introducing claude sonnet 4.5, 2025. URL <https://www.anthropic.com/news/claude-sonnet-4-5>.
- 546
547
- 548 Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with gemini 3, 2025. URL <https://blog.google/products-and-platforms/products/gemini/gemini-3/#gemini-3>.
- 549
550
551
- 552 Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize
553 Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chat-
554 Dev: Communicative agents for software development. In Lun-Wei Ku, Andre Martins, and
555 Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, Bangkok, Thailand, August
556 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.810. URL
557 <https://aclanthology.org/2024.acl-long.810/>.
- 558
- 559 Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan
560 McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar,
561 Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel,
562 Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer,
563 Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa
564 Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone,
565 Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt,
566 Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie
567 Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra,
568 Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak,
569 Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang,
570 Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner,
571 Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson,
572 Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen,
573 Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina
574 Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory
575 Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts,
576 Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson,
577 Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras,
578 Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin
579 Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung,
580 Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin
581 Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila
582 Belbute Peres, Filippo Raso, Florent Bekermer, Foivos Tsimpourlas, Fotis Chantzis, Francis Song,
583 Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo,
584 Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson
585 Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt,
586 Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian
587 Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina
588 Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski,
589 Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason
590 Wang, Jason Wolfe, Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao,
591 Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin,
592 Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes
593 Ferstod, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan
Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh
McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen,
Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal,
Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu,

- 594 Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry
595 Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah
596 Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang
597 Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan
598 Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz
599 Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark
600 Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeih, Matt Nichols,
601 Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Huang, Meng Dong,
602 Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman,
603 Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang,
604 Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa
605 Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu,
606 Nick Preston, Nick Turley, Nicklas Ansmann, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko
607 Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg
608 Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel
609 Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua
610 Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara,
611 Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar,
612 Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan
613 Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel
614 Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean
615 Taylor, Sebastien Bubeck, Sever Banerjee, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani
616 Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang
617 Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song
618 Mei, Sonya Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe,
619 Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer,
620 Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas
621 Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki
622 Sherbakov, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Pe-
623 terson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek
624 Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes
625 McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu
626 Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian,
627 Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai,
628 Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun
629 Wang, Yushi Wang, Yutian Liu, Zach Stuebenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang.
630 Openai gpt-5 system card, 2025. URL <https://arxiv.org/abs/2601.03267>.
- 629 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally
630 can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- 632 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le,
633 and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
634 URL <https://arxiv.org/abs/2201.11903>.
- 636 Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun
637 Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and
638 Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023. URL
639 <https://arxiv.org/abs/2308.08155>.
- 640 Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An
641 empirical analysis of compute-optimal inference for problem-solving with language models, 2025.
642 URL <https://arxiv.org/abs/2408.00724>.
- 644 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
645 Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
646 Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
647 Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,
Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui

648 Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang
649 Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger
650 Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan
651 Qiu. Qwen3 technical report, 2025a. URL <https://arxiv.org/abs/2505.09388>.
652

653 Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang.
654 Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems, 2025b. URL
655 <https://arxiv.org/abs/2504.00587>.

656 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik
657 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.
658 URL <https://arxiv.org/abs/2305.10601>.

659 Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arık. Chain of
660 agents: Large language models collaborating on long-context tasks. In A. Globerson, L. Mackey,
661 D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information*
662 *Processing Systems*, volume 37, pp. 132208–132237. Curran Associates, Inc., 2024. doi: 10.52202/
663 079017-4202. URL [https://proceedings.neurips.cc/paper_files/paper/
664 2024/file/ee71a4b14ec26710b39ee6be113d7750-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2024/file/ee71a4b14ec26710b39ee6be113d7750-Paper-Conference.pdf).
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

A EXPERIMENT SETUP DETAILS

A.1 ADDITIONAL EXPERIMENT SETUP

For reasoning-intensive models, we applied the default "medium" effort setting, capping the reasoning token budget at 50% of the maximum generation length to ensure consistent comparisons across all baselines.

For the LiveCodeBench benchmark, we use the data introduced by the newest released version v6 (i.e. data from 2025/01/01 to 2025/05/01).

In the orchestration agent selection experiment, we judge the performance of the orchestrator by activating all tool agents. The max generation token is set based on each agent's cost, ensuring it will not exceed the cost budgets.

A.2 AGENTVERSE SETUP

We employ GPT-5-Mini (Singh et al., 2025) as the backbone language model of agents in AgentVerse (Chen et al., 2024). We use a maximum token limit of 512 for the role assigner agent, and 4,096 for the rest of the agents. For invalid agent outputs, such as invalid role assignments, unparseable answers, or errors in code, AgentVerse retries generation a limited number of times: 10 times on math tasks and 1,000 times on coding tasks.

B PERFORMANCE ACROSS THE NUMBER OF ACTIVE TOOL AGENTS

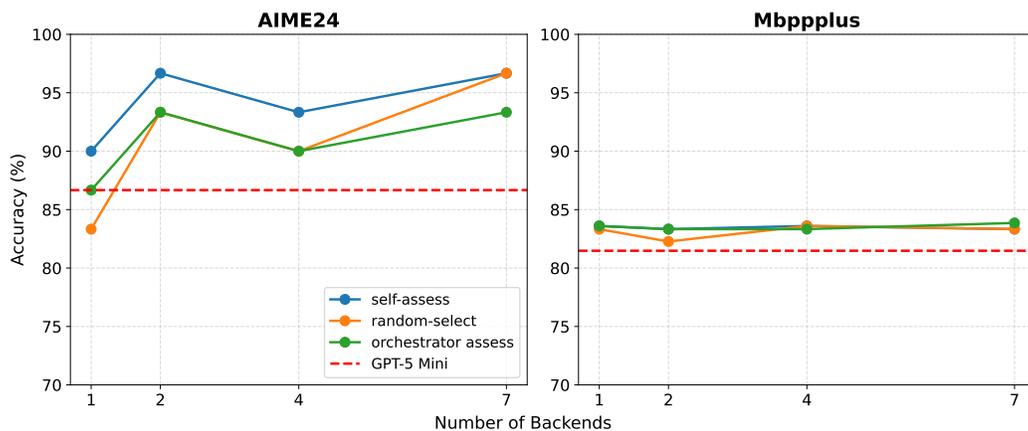


Figure 3: Comparison of different tool agent-selection methods in AIME2024 and MBPP+, aggregated using GPT-5-mini as the orchestrator. Dashed lines indicate baseline GPT-5-mini performance.

We further analyze how the number of activated tool agents affects aggregation performance. Figure 3 shows that the performance of the three selection methods converges as the number of active tools increases. On AIME, the performance gap between selection strategies is more pronounced, whereas on MBPP+ the differences are substantially smaller.

C EXPERIMENT PROMPT

Here is an example of the language model-based tool agent assessment prompts.

Assessment Prompt

Instructions: You will be provided with a series of problems. For each problem, the Subject Agent has provided an answer. Some are correct, and some are incorrect.

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

Part 1: Per-Problem Analysis

For every problem provided, generate a structured audit containing:

1. **Taxonomy:** Classify the problem type (e.g., Arithmetic, Logical Reasoning, Creative Writing, Coding) and specific skill required.
2. **Performance Verdict:** Clearly state [PASS] or [FAIL].
3. **Gap Analysis:**
 - *If Correct:* Briefly explain why the agent succeeded (e.g., “Good step-by-step reasoning,” “Robust knowledge retrieval”).
 - *If Incorrect:* Pinpoint exactly *where* and *why* the agent failed. Was it a calculation error? A logic jump? A hallucination? A misunderstanding of constraints? Compare the Subject’s logic to the Ground Truth.

Part 2: Executive Summary

After analyzing all problems, synthesize a “Model Persona” profile:

1. **Core Competencies:** List specific categories where the agent consistently succeeds.
2. **Blind Spots & Failure Modes:** Describe the patterns in the agent’s errors (e.g., “The agent struggles with negative integers,” or “The agent is verbose but inaccurate”).
3. **Final Verdict:** A 2-sentence summary of the agent’s reliability.

—
Input Data:

- Problem 1: *Question, Subject Agent Answer, Ground Truth/Solution...*
- Problem 2: *...repeat for all samples...*

—
COMMAND:

Based on the data above, proceed with the **Per-Problem Analysis** followed by the **Executive Summary**.

Reminders:

1. **Be Specific:** Do not just say “The agent failed.” Identify *why* (e.g., Logic Error vs. Calculation Error).
2. **Be Critical:** Compare the Subject Answer against the Ground Truth rigorously.
3. **Format:** Use the headers ## Part 1: Per-Problem Analysis and ## Part 2: Executive Summary.

GENERATE REPORT: