Two Heads are Better Than One: Test-time Scaling of Multi-agent Collaborative Reasoning

Can Jin[1*], Hongwu Peng[2], Qixin Zhang[3], Yang Zhou[1], Yujin Tang[4]
Tong Che[5†], Dimitris N. Metaxas[1†]
[1]Rutgers University [2]University of Connecticut [3]Nanyang Technological University
[4]Sakana AI [5]NVIDIA Research
[*]Correspondence to: can.jin@rutgers.edu [†]Equal Advising

Abstract

Test-Time Scaling has emerged as a powerful method to extend the reasoning capabilities of Large Language Models. However, single-agent TTS faces significant scalability bottlenecks, as excessively long reasoning traces lead to increased inference costs and stability issues caused by context management failures. To address these limitations, we propose leveraging Multi-Agent Systems as a structural upgrade to standard TTS. By decomposing monolithic reasoning chains into distinct, manageable contexts across multiple agents, MAS offers a more robust framework for scaling reasoning. We validate this approach by introducing M500, a dataset comprising 500 high-quality multi-agent, multi-turn collaborative reasoning traces generated via DeepSeek-R1. Through Supervised Fine-Tuning on M500, we enable open-source models to internalize collaborative reasoning patterns and show improved TTS performance in MAS. Furthermore, we propose an adaptive scaling strategy incorporating a "CEO" agent to dynamically guide the reasoning process and optimize collaboration depth. Extensive experiments within the AgentVerse framework demonstrate that our fine-tuned models, Qwen2.5-32B-MAS and Phi4-14B-MAS, significantly outperform their base counterparts. Codes are available at https://github.com/jincan333/MAS-TTS.

1 Introduction

The advancement of Large Language Models (LLMs) (Achiam et al., 2023; Hurst et al., 2024; Liu et al., 2024a; Yang et al., 2024; Abdin et al., 2024; Jin et al., 2024) has significantly extended the capabilities of artificial intelligence. Recently, Test-Time Scaling (TTS) has emerged as an effective approach to further enhance LLM performance, particularly for complex mathematical reasoning tasks (Guo et al., 2025; Snell et al., 2025; Liu et al., 2025; Muennighoff et al., 2025; Wu et al., 2025; Jin et al., 2026). Techniques such as Monte Carlo Tree Search (Gao et al., 2024; Zhang et al., 2024b), large-scale reinforcement learning (Guo et al., 2025; OpenAI, 2024; Zeng et al., 2024), and supervised fine-tuning (SFT) on detailed reasoning chains (Muennighoff et al., 2025; Ye et al., 2025) have been widely applied to extend the inference process. However, extending the reasoning trace for a single agent introduces significant challenges. As the reasoning chain grows excessively long, the inference cost increases substantially(Dao et al., 2022), and the model becomes prone to divergence and hallucination due to the difficulty of managing a massive context window effectively(Liu et al., 2024b; Dziri et al., 2023; Turpin et al., 2023).

To address the limitations of single-agent TTS, we propose leveraging Multi-Agent Systems (MAS) as a structural extension to standard TTS. The core advantage of MAS in this context is the ability to decompose a potentially unstable, monolithic reasoning trace into distinct segments managed by different roles (Feng et al., 2025; Li et al., 2023). By assigning specific contexts and sub-tasks to different agents, MAS can maintain coherence over long-form reasoning processes that would otherwise overwhelm a single agent (Chen et al., 2024; Qian et al., 2024; Schmidgall et al., 2025; Lu et al., 2024). Consequently, MAS serves as a more robust framework for TTS, allowing for effective scaling of reasoning capabilities without the context management failures observed in single-agent settings.

Building on the concept of Simple-Scaling (Muennighoff et al., 2025), we hypothesize that the collaborative reasoning capabilities required for this enhanced TTS can be established via SFT on multi-agent Chain-of-Thought (CoT) interaction data. To investigate this, we first curate a high-quality multi-agent collaborative reasoning dataset by solving a diverse set of challenging problems using an open-source MAS. Each instance in the dataset comprises the

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 8984–9004
October 24-29, 2026 ©2026 Association for Computational Linguistics

interactive sequence of multi-agent, multi-turn data generated by the MAS to solve a specific problem. To ensure dataset quality and support long-form CoT, we filter low-quality examples and use DeepSeek-R1 (Guo et al., 2025) to generate robust reasoning traces for every agent. We designate this dataset as M500, which contains 500 diverse and detailed multi-agent collaborative reasoning traces. We then perform SFT on M500 using standard open-source, instruction-tuned LLMs. To further optimize adaptive reasoning and scaling in the MAS, we introduce a "**CEO**" agent powered by the SFT model, inspired by the observation that leaderless groups in human societies often lack effectiveness and coherent direction (Cruz et al., 1999; Hackman, 2002). This agent dynamically guides discussions, effectively managing collaborative efforts and reasoning depth to improve the overall performance of the system.

We conduct extensive experiments to validate our approach by fine-tuning Qwen2.5-32B-Instruct (Hui et al., 2024) and Phi4-14B (Abdin et al., 2024) on M500, yielding the models Qwen2.5-32B-MAS and Phi4-14B-MAS. We then integrate these models into the AgentVerse (Chen et al., 2024) multi-agent framework. Testing across various task categories—including general understanding, mathematical reasoning, and coding—demonstrates that our Qwen2.5-32B-MAS and Phi4-14B-MAS significantly surpass the baseline Qwen2.5-32B-Instruct and Phi4-14B within the MAS. For example, our method achieves a 12% improvement on GPQA-Diamond (Rein et al., 2024), a 41% improvement on AIME2024 (MAA, 2024), and a 10% improvement on MBPP-Sanitized (Austin et al., 2021), while matching the performance of DeepSeek-R1 on MATH-500 and MBPP-Sanitized.

In summary, our contributions are: ❶ We develop M500, a comprehensive multi-agent, multi-turn collaborative reasoning dataset created via an automatic generation pipeline; ❷ We demonstrate the collaborative reasoning capability of LLMs in MAS can be effectively improved by SFT on multi-agent, multi-turn reasoning datasets, serving as an effective implementation of MAS-TTS; ❸ We propose an adaptive scaling strategy that incorporates a CEO agent powered by the obtained LLMs to dynamically guide and scale multi-agent collaborative reasoning; and ❹ We show through extensive experiments that our method significantly outperforms baseline models and achieves performance comparable to DeepSeek-R1 on specific tasks.

2 Related Works

2.1 Test-time Scaling

A wide range of methods have been developed to improve reasoning in LLMs by leveraging Test-Time Scaling (TTS). Recent work explores techniques including hierarchical hypothesis search, which enables inductive reasoning through structured exploration (Wang et al., 2023b; Li et al., 2026). They also examine tool augmented inference, which enhances downstream performance by allowing models to interact with external environments (Gao et al., 2023; Qin et al., 2024). Other approaches focus on internal mechanisms, such as learning thought tokens in an unsupervised manner (Zelikman et al., 2024; Goyal et al., 2023), allowing models to better utilize extended reasoning sequences. Among the most studied scaling paradigms are parallel and sequential TTS approaches. Parallel methods generate multiple solution candidates independently and select the best one using a scoring criterion, such as majority voting or outcome-based reward models (Brown et al., 2024; Irvine et al., 2023; Snell et al., 2025). In contrast, sequential methods condition each new attempt on the previous ones, allowing iterative refinement based on prior outputs (Muennighoff et al., 2025; Snell et al., 2025; Hou et al., 2025; Lee et al., 2025). Bridging these strategies, tree-based techniques such as Monte Carlo Tree Search (MCTS) (Zhang et al., 2023; Zhou et al., 2023; Zhang et al., 2024a; Wu et al., 2025) and guided beam search (Xie et al., 2023) enable structured exploration through branching and evaluation. Central to many of these methods are reward models, which provide feedback signals during generation. These can be categorized as outcome reward models, which evaluate entire solutions (Xin et al., 2024; Ankner et al., 2024), or process reward models, which assess intermediate reasoning steps (Lightman et al., 2023; Wang et al., 2023a; Wu et al., 2025), guiding the model toward more effective reasoning paths.

2.2 LLM Agents

Recent work has extended the capabilities of LLMs beyond standalone reasoning and understanding, enabling them to operate as multi-agents that can interact with environments, tools, and other agents to perform complex tasks (Wu et al., 2023; Li et al.,

2023; Chen et al., 2024; Qian et al., 2024; Motwani et al., 2024; Jin et al., 2025a,b). These multi-agent systems (MAS) integrate various techniques, including CoT prompting (Wei et al., 2022), iterative refinement (Shinn et al., 2023), self-improvement (Huang et al., 2023), and external tool usage (Hao et al., 2023; Qin et al., 2024; Schick et al., 2023), to support multi-step decision-making and long-horizon planning. They have been applied successfully in domains such as mathematical reasoning (Chen et al., 2024), software engineering (Qian et al., 2024; Jimenez et al., 2023; Wang et al., 2024), and scientific discovery (Lu et al., 2024; Schmidgall et al., 2025). Agent frameworks typically structure the interaction with LLMs using techniques such as few-shot prompting (Brown et al., 2020) and guided reasoning (Wei et al., 2022; Shinn et al., 2023), relying on the model's in-context learning capabilities (Olsson et al., 2022).

3 Methodology

We first describe the automatic generation of high-quality multi-agent collaborative reasoning data. Next, we improve the collaborative reasoning capabilities of LLMs by performing SFT on the generated data. Finally, we introduce a CEO agent into the MAS to further enable adaptive scaling by directing collaboration and adjusting resource allocation. Our method overview is in Figure 3.

3.1 Automatic Generation of Multi-Agent Collaborative Reasoning Data

Question Sampling Based on Difficulty, Diversity, and Interdisciplinarity. When selecting questions for our multi-agent collaborative reasoning dataset, we consider three main aspects: ❶ Difficulty, ❷ Diversity, and ❸ Interdisciplinarity. We begin with the full Simple-Scaling dataset (Muennighoff et al., 2025), which includes diverse questions sourced from historical AIME problems, OlympicArena (Huang et al., 2024), and AGIEval (Zhong et al., 2024), among others. These questions span multiple domains, including physics, geometry, number theory, biology, and astronomy. To promote difficulty and interdisciplinarity, we use DeepSeek-R1 (Guo et al., 2025) to determine whether solving each question requires interdisciplinary knowledge, excluding those that DeepSeek-R1 answers using fewer than 1024 tokens. Questions selected through this process are generally challenging and demand knowledge integration from multiple disciplines. For example, solving a complex mathematics problem might benefit from collaboration between algebra and geometry specialists, whereas an advanced astronomy question may require input from astronomers, physicists, and mathematicians.

Multi-agent Collaborative Reasoning Trace

Expert Recruiter
Prompt: You are leading a group of experts to solve: ... Find the greatest integer that does not exceed $\frac{x^2}{100}$. Recruit two experts with different specialties to best solve the problem...
Response: <think> ... I need to identify experts best suited for this geometry task... </think>1. A geometric analysis expert. 2. An algebraic modeling expert.

Problem Solver 1 (Geometric Expert)
Prompt: ... Solve the following accurately: One base of a trapezoid is 100 units longer than the other... Find the greatest integer that does not exceed $x^2/100$...
Response: <think> ... Let's understand the structure: the segment splits the area in a 2:3 ratio. We need the length x such that both halves have equal area... </think>...Using RMS to find x, we get:

$$\frac{x^2}{100} = \frac{18125}{100} = 181.25 \Rightarrow \boxed{181}$$

Problem Solver 2 (Algebraic Expert)
Prompt: ... Given the prior solution, critique its correctness. Do not solve again. If you agree with the final answer, append `[Agree]`...
Response: <think> ... Reviewing the solution: the answer is correct. However, explanation for using RMS could be clearer... </think>... The reasoning is sound and the steps are accurate. `[Agree]`

Evaluator
Prompt: You are an experienced science teacher. Evaluate the final solution from both solvers for correctness...
Response: <think> ... Let me verify each step. Yes, the use of RMS and the calculation are both appropriate... </think>... **Correctness:** 1 ...

Figure 1: An illustration of a multi-agent collaborative reasoning data sample from M500.

Generating Multi-Agent Collaborative Reasoning Traces. To generate collaborative reasoning traces, we employ the open-source MAS frameworks and reasoning models, specifically AgentVerse (Chen et al., 2024) and DeepSeek-R1 (Guo et al., 2025), to process previously selected questions. This MAS framework involves multiple roles: Expert Recruiter (e.g., Human Resources), Problem Solvers (e.g., Scientists and Engineers), Executor (e.g., Quality Assurance Engineers), and Evaluator (e.g., Product Manager). The agents collaborate in the following manner: the Expert Recruiter identifies and assigns suitable experts for the task, while keeping the **total agent number** fixed and independent of question complexity or available information. These recruited experts function as Problem Solvers, proposing and iteratively refining solutions through multi-turn discussions with a predetermined **number of critic iterations**. Once

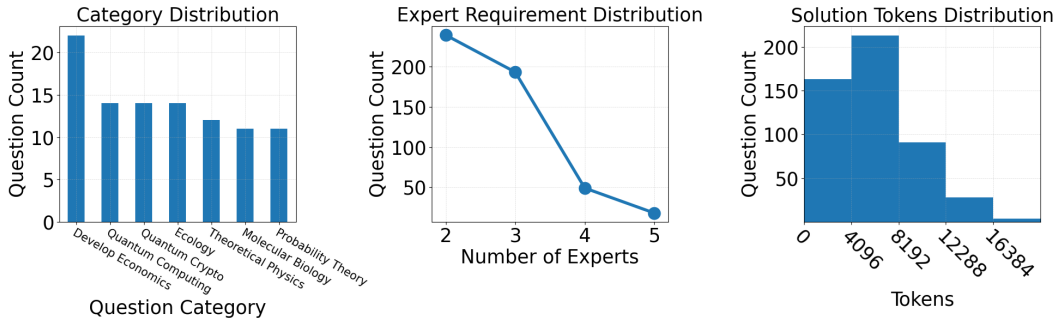

Figure 2: Distributions of key statistics in M500: question category (filtered with count > 10), predicted number of experts required for solving each problem, and solution token usage.

a consensus is reached, or if the maximum number of critic iterations is reached, the resulting solution is passed to the Executor, who runs the necessary code or invokes external tools if required. The Evaluator then reviews both the solution and the results produced by the Executor, providing feedback for further refinement. This iterative process continues until either the solution is deemed correct or the predefined **maximum total number of iterations** is reached. Each MAS role produces distinct inputs and outputs, generating reasoning traces from its expert viewpoint. We collect all input-output pairs from each agent and compile the entire collaborative reasoning history into a comprehensive trace for each question.

Data Filtering. To ensure high-quality collaborative reasoning traces, we filter data based on three criteria: ❶ **Consensus Reached:** Traces where Problem Solvers fail to reach consensus within the maximum allowed critic iterations are discarded. This criterion ensures effective discussion convergence and minimizes unresolved reasoning. ❷ **Format Compliance:** Samples that deviate from the required format are excluded. Specifically, each agent's reasoning must be enclosed within `<think>` and `</think>` tags, and the final answers must be encapsulated within `boxed{}`. This maintains output consistency and facilitates automated parsing and answer extraction. ❸ **Correctness:** We extract the final answer from each collaborative trace and compare it against the ground-truth solution, discarding any traces with incorrect answers.

Through this method, we iteratively sample 500 questions along with their corresponding multi-agent collaborative reasoning traces, forming the **M500** dataset. This dataset contains 500 challenging and diverse questions requiring interdisciplinary collaboration, accompanied by comprehensive and high-quality reasoning traces that document the complete multi-agent problem-solving process. The pseudocode detailing this sampling procedure is provided in Algorithm 1 in the Appendix.

Figure 1 shows an illustrative example from M500, with the complete data sample presented in Figure 7 in the Appendix. The collaborative reasoning trace for this example includes input-output interactions among four agents: Expert Recruiter, Geometry Expert, Algebra Expert, and Evaluator. The example question is sufficiently challenging (requiring 5695 tokens), achieves consensus among agents, complies with the required format, and produces a correct solution. Additionally, the distribution of question categories in the M500 dataset, predicted expert counts, and solution token usage is illustrated in Figure 2. We observe significant diversity in the dataset across fields such as economics, physics, biology, and mathematics. Most questions are predicted to be solved by two experts and require fewer than 8192 tokens to solve.

3.2 Enhancing LLM Collaborative Reasoning through Supervised Fine-Tuning

Inspired by Simple-Scaling (Muennighoff et al., 2025), which shows that long-form CoT reasoning capabilities in LLMs can be developed through SFT on detailed reasoning traces, we apply SFT to an LLM f using the M500 dataset. The goal is to enable f to produce long-form CoT reasoning that supports the collaboration in a MAS. Specifically, the SFT objective is to minimize the following loss:

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}}[-\tfrac{1}{|\mathbf{y}|}\textstyle\sum_{t=1}^{|\mathbf{y}|}\log P_f(\mathbf{y}_t \mid \mathbf{x}, \mathbf{y}_{<t})]$$

where $P_f(\mathbf{y}_t \mid \mathbf{x}, \mathbf{y}_{<t})$ denotes the probability the model f assigns to token $\mathbf{y}_t$ given input $\mathbf{x}$ and previous tokens $\mathbf{y}_{<t}$.

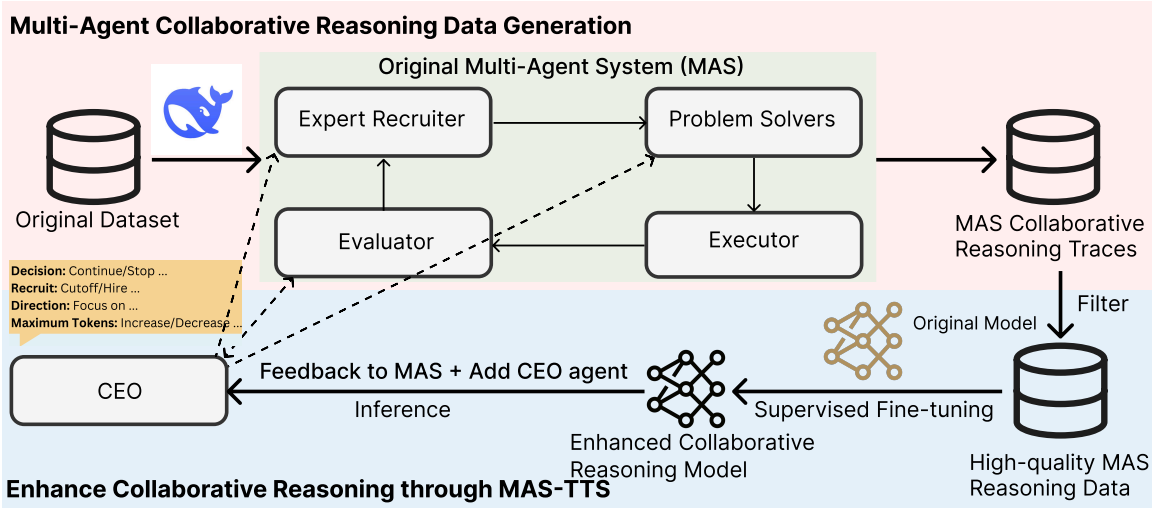

Figure 3: Overview of our method. First, we generate high-quality collaborative reasoning traces (M500) using the AgentVerse (Chen et al., 2024) framework. We then perform SFT on the base model using M500 to improve its collaborative reasoning capabilities. Finally, we integrate the fine-tuned model back into the MAS and introduce a CEO agent to adaptively scale reasoning by dynamically adjusting collaboration intensity and reasoning depth.

For each question q in the M500 dataset, we collect a series of input-output pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ corresponding to the reasoning traces from all participating agents. During training, we ensure all reasoning traces for q, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, are grouped within the same batch and ordered according to the original generation sequence in the MAS. This approach helps the model learn collaborative reasoning in a coherent and temporally consistent manner.

3.3 Adaptive Test-time Scaling

Recently, TTS has emerged as an effective method for enhancing the performance of LLMs. Models such as OpenAI's o-series and DeepSeek-R1 have shown considerable improvements by employing scaled reasoning during inference. However, the application of TTS within MAS remains relatively unexplored. Previous studies in single-agent scenarios indicate that the optimal TTS strategy depends on question difficulty (Liu et al., 2025; Wu et al., 2025). In MAS, choosing an appropriate TTS strategy is even more critical due to the significantly higher computational and time costs involved in collaboration compared to single-agent.

To address this issue, we propose an adaptive TTS strategy for MAS by introducing a dedicated "**CEO**" agent, which dynamically manages collaboration and resource allocation based on the task's ongoing progress. As shown in Figure 3, the CEO agent evaluates the question, current solution state, evaluation feedback, and available resources to determine whether a proposed solution should be accepted or needs further refinement. Additionally, this agent directs subsequent discussions, decides how many agents to involve, and sets appropriate reasoning depth (i.e., the token budget for each agent's response).

Unlike static MAS configurations, which have fixed numbers of agents, iteration limits, and reasoning depths, our adaptive approach allows the MAS to dynamically adjust its settings. This capability enables more effective scaling of collaborative reasoning by modifying agent participation, termination conditions, and reasoning depth according to the evolving requirements of the problem.

4 Experiments

To validate that our method effectively improves collaborative reasoning in MAS, we conduct experiments using state-of-the-art (SOTA) open-source and closed-source LLMs within the AgentVerse framework. We evaluate performance across general understanding, mathematical reasoning, and coding tasks. Additionally, we analyze the emergent behavior and scaling properties of the proposed method.

4.1 Experimental Details

LLMs. We evaluate both reasoning and non-reasoning LLMs to assess the impact of collaborative reasoning in MAS. Our primary baselines are Qwen2.5-32B-Instruct (Hui et al., 2024; Yang et al., 2025) and Phi4-14B (Abdin et al., 2024). To isolate the benefits of MAS-specific SFT, we com-

Model	General Understanding		Mathematical Reasoning		Coding		
	GPQA	Commongen	AIME2024	MATH-500	HumanEval	MBPP-S	Avg.
API-accessed Models							
DeepSeek-V3	58.6	98.6	33.3	88.6	89.6	83.9	75.4
GPT-4o	49.2	97.8	7.8	81.3	90.9	85.4	68.7
DeepSeek-R1	**75.5**	97.2	78.9	**96.2**	**98.2**	91.7	89.6
o3-mini	71.3	**99.1**	**84.4**	95.3	97.0	**93.6**	**90.1**
Locally deployed models							
Phi4-14B	53.5	95.4	13.3	80.6	83.2	79.1	67.5
Phi4-14B-SFT	56.3	95.2	50.0	84.3	85.4	81.8	75.5
Qwen2.5-32B-Instruct	50.2	96.7	21.1	84.4	89.0	80.2	70.3
Qwen2.5-32B-SFT	56.1	96.2	53.3	88.6	85.2	84.1	77.3
s1.1-32B	58.3	94.1	53.3	90.6	82.3	77.4	76.0
Phi4-14B-MAS (Ours)	58.1	95.8	56.7	88.2	85.8	83.2	78.0
Phi4-14B-MAS w. CEO (Ours)	58.8	96.2	60.0	89.8	87.5	84.6	79.5
Qwen2.5-32B-MAS (Ours)	61.1	96.9	60.0	95.1	92.8	89.1	82.5
Qwen2.5-32B-MAS w. CEO (Ours)	**62.1**	**97.4**	**62.2**	**95.8**	**93.9**	**90.5**	**83.7**

Table 1: Performance comparison on general understanding, mathematical reasoning, and coding tasks using strong reasoning and non-reasoning models within the AgentVerse framework. Our method achieves substantial improvements over baselines on all tasks and attains performance comparable to o3-mini and DeepSeek-R1 on MATH-500 and MBPP-S, demonstrating its effectiveness in enhancing collaborative reasoning in MAS.

pare our models against single-agent SFT baselines: Qwen2.5-32B-SFT and Phi4-14B-SFT. These baselines are fine-tuned on the same problem set as M500 but utilize direct, single-agent solutions generated by DeepSeek-R1. Additionally, we include s1.1-32B, which is trained on questions from Simple-Scaling (Muennighoff et al., 2025) using DeepSeek-R1 generated responses. Our models, Qwen2.5-32B-MAS and Phi4-14B-MAS, along with their respective SFT baselines, are fine-tuned from Qwen2.5-32B-Instruct and Phi4-14B, respectively. We also benchmark against strong open-source models, DeepSeek-V3 (Liu et al., 2024a) and DeepSeek-R1 (Guo et al., 2025), as well as closed-source models, including o3-mini (medium) (OpenAI, 2025) and GPT-4o (2024-08-06) (Hurst et al., 2024).

Tasks. We conduct a comprehensive evaluation across three critical domains: ❶ **General Understanding**: We employ GPQA-Diamond (GPQA) (Rein et al., 2024) to assess general knowledge and Commongen-Challenge (Commongen) (Madaan et al., 2023) to evaluate writing capability. GPQA comprises 198 PhD-level science questions in Biology, Chemistry, and Physics; We report zero-shot accuracy. Commongen requires agents to generate coherent paragraphs incorporating 20 specific concepts. For Commongen, we report the average percentage of concepts covered across 200 test cases. ❷ **Mathematical Reasoning**: We utilize two challenging benchmarks: AIME2024 (MAA, 2024) and MATH-500 (Hendrycks et al., 2021). AIME2024 consists of 30 problems from the 2024 American Invitational Mathematics Examination, while MATH-500 contains competition-level math problems of varying difficulty. We report zero-shot accuracy for both. ❸ **Coding**: We assess code generation capabilities using HumanEval (Chen et al., 2021) and MBPP-Sanitized (MBPP-S) (Austin et al., 2021). HumanEval includes 164 Python programming problems, and MBPP-S contains 257 introductory Python problems. We report zero-shot Pass@1 accuracy for these tasks.

Training and Evaluation. We perform SFT on the M500 dataset for 5 epochs with a learning rate of 1×10^{-5} to obtain Qwen2.5-32B-MAS and Phi4-14B-MAS. To ensure a fair comparison, Qwen2.5-32B-SFT and Phi4-14B-SFT are trained using identical hyperparameters. Training is conducted on 8 NVIDIA A100 (80GB) GPUs using FlashAttention (Dao, 2024) and DeepSpeed (Rasley et al., 2020) within the LLaMA-Factory framework (Zheng et al., 2024). Evaluation is performed using the AgentVerse (Chen et al., 2024) framework, configured with 5 agents, 3 critic iterations, and 2 total iterations by default. The final output generated by the MAS is used for metric calculation. All primary results are averaged over

three independent runs. Prompts for mathematical reasoning tasks are detailed in Appendix B, while prompts for other tasks are available in the accompanying code. The M500 dataset and source code are provided in the supplementary materials.

4.2 Main Results

We demonstrate that our method significantly improves the performance of LLMs within MAS compared to both the original models and SFT models trained on single-agent datasets. The experimental results are presented in Table 1. We observe several key findings: ❶ Our method achieves substantial improvements across all evaluated tasks relative to Qwen2.5-32B-Instruct and Phi4-14B. This demonstrates that Qwen2.5-32B-MAS and Phi4-14B-MAS, particularly when integrated with the CEO agent, effectively improve general question answering, writing, mathematical reasoning, and coding capabilities within MAS. Specifically, Qwen2.5-32B-MAS w. CEO improves performance by 12%, 41%, 11%, and 10% on GPQA, AIME2024, MATH-500, and MBPP-S, respectively, compared to Qwen2.5-32B-Instruct. Moreover, our method achieves performance comparable to SoTA open-source and closed-source models, such as DeepSeek-R1 and o3-mini, on MATH-500, Commongen, and MBPP-S, underscoring the effectiveness of our approach. ❷ Our approach significantly improves collaborative reasoning in MAS compared to single-agent SFT and Simple-Scaling (Muennighoff et al., 2025). For instance, Qwen2.5-32B-MAS with CEO outperforms s1.1-32B by 4% on GPQA and 9% on AIME2024. Additionally, while s1.1-32B experiences performance degradation in coding tasks compared to Qwen2.5-32B-Instruct, our method yields notable improvements in this domain, highlighting the advantage over Simple-Scaling. ❸ The MAS models outperform the single-agent SFT models when evaluated within the MAS framework, indicating that collaborative reasoning traces are more effective than single-agent reasoning data for enhancing multi-agent capabilities. ❹ The introduction of the CEO agent consistently improves the performance of Qwen2.5-32B-MAS and Phi4-14B-MAS across all tasks. This highlights that collaborative reasoning in MAS is effectively scaled when guided by a CEO agent powered by Qwen2.5-32B-MAS or Phi4-14B-MAS.

Aha Moment in MAS

Question
Let ABC be a triangle inscribed in circle ω. The tangents to ω at B and C intersect at point D, and $\overline{AD}$ intersects ω again at P. If $AB = 5$, $BC = 9$, and $AC = 10$, AP can be written as $\frac{m}{n}$ in lowest terms. Find $m + n$.

Problem Solver
... Place $B = (0, 0)$, $C = (9, 0)$, compute power of D: $DB^2 = \left(\frac{225}{22}\right)^2$. Use the power of a point: $DA \cdot DP = DB^2$, solve for DP, then compute $AP = DP - DA$. Final result: $AP = \frac{22}{1} \Rightarrow m + n = \boxed{23}$.

CEO
<think>...Wait, let's see what the solution did. They said $DA = \frac{325}{22}$, $DB^2 = \left(\frac{225}{22}\right)^2$. Then:

$$DP = \frac{(225/22)^2}{325/22}, \quad AP = DP - DA = \frac{2025}{286} - \frac{325}{22} = \frac{-1100}{143}$$

But length can't be negative, which indicates a mistake in the sign. This suggests that the assumption about the position of P is wrong. Maybe P is between D and A. So $AP = \frac{1100}{143} = \frac{100}{13} \approx 7.6923$
The solution says $AP = 22$. That's impossible. The correct value is $AP = \frac{100}{13}$, hence $m + n = \boxed{113}$... </think>
Decision: **Continue**
Recruit Number: 2
Direction: 1) Verifying the power of point application 2) Re-examining the geometric configuration ...
Maximum Tokens: 8192

Problem Solver
... Power of point D calculation: $DA \cdot DP = DB^2$
Where: $DA = \frac{325}{22}$ (distance from D to A), $DB = \frac{225}{22}$ (tangent length from D to circle). Solving for DP:

$$DP = \frac{DB^2}{DA} = \frac{\left(\frac{225}{22}\right)^2}{\frac{325}{22}} = \frac{2025}{286}$$

Critical geometric relationship: Point P lies between D and A, so: $AP = DA - DP = \frac{325}{22} - \frac{2025}{286} = \frac{100}{13}$
Final result: $AP = \frac{100}{13} \Rightarrow m + n = \boxed{113}$

Figure 4: An "aha" moment in MAS where the CEO agent proactively verifies and corrects the solution provided by the Problem Solver. After identifying an error, the CEO suggests a corrected approach, which the Problem Solver then incorporates into its revised solution.

4.3 Aha Moment in MAS

In MAS, we observe that when using Qwen2.5-32B-MAS, agents sometimes exhibit emergent behaviors that actively contribute to validating and refining collaborative processes, even when it is not explicitly required. For example, as illustrated in Figure 4, the Problem Solver initially fails to recognize an error in its solution. After reviewing this solution, the CEO agent actively checks its validity and identifies the Problem Solver's error, noting that it results in a negative length for a line segment. The CEO agent then proposes an alternative and correct solution, prompting the Problem Solver to revise its original response accordingly. This collaborative interaction, where one agent assists

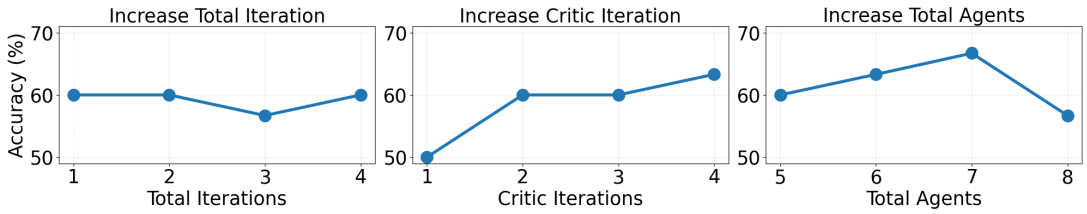

Figure 5: The effect of scale collaboration in AgentVerse using Qwen2.5-32B-MAS by increasing the total iteration, critic iteration, and total agents involved in the MAS.

others by verifying solutions, exploring alternative approaches, and suggesting corrections, occurs even when other agents are unaware of their own mistakes. A plausible reason for this emergent behavior is that the CEO agent, having been trained on multi-agent collaborative reasoning traces and observing other agents' discussions, actively validates and corrects solutions based on learned collaborative patterns and insights gained from the reasoning of other agents.

4.4 Additional Investigation

Scaling Collaborative Reasoning in MAS. We investigate how scaling collaborative reasoning affects the performance of Qwen2.5-32B-MAS in MAS by systematically adjusting the total iterations, critic iterations, total agent numbers, and maximum token limits. The results are presented in Figures 5 and 6. Our observations are as follows: ❶ Enhancing collaboration by increasing the interactions among Problem Solvers significantly improves performance. This can be achieved either by raising the critic iteration limit to allow more extensive discussion toward consensus or by increasing the total number of Problem Solvers. However, involving too many Problem Solvers may reduce performance due to divergent discussions among agents. Additionally, merely increasing the total iterations does not improve MAS performance. ❷ Enhancing reasoning capabilities by increasing the maximum allowed tokens per agent effectively improves MAS performance. Furthermore, optimal token limits vary by task; for example, 16384 tokens yield optimal results for AIME2024, whereas 8192 tokens are sufficient for GPQA. This finding supports our motivation for using the CEO agent to dynamically manage token allocation based on specific task requirements.

Performance of Qwen2.5-32B-MAS as a Single Agent. We further investigate the performance improvement achieved by using Qwen2.5-32B-MAS within MAS compared to its performance as a single agent setting. The results are summarized in Table 2. We observe that employing Qwen2.5-32B-MAS in MAS significantly improves performance compared to its single-agent performance. In contrast, using Qwen2.5-32B-Instruct and SFT models within MAS results in smaller improvements over the single-agent setting, further demonstrating the effectiveness of our proposed method in enhancing MAS performance.

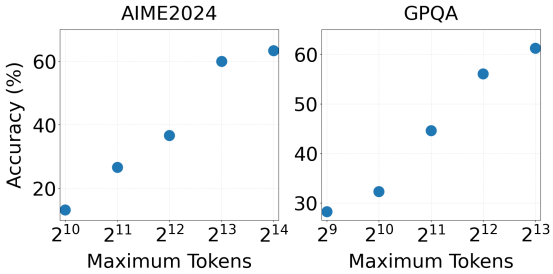

Figure 6: Effect of scaling reasoning on AgentVerse using Qwen2.5-32B-MAS by controlling the token usage.

Models	AIME2024	GPQA
Qwen2.5-32B-Instruct + SA	26.7	49.0
Qwen2.5-32B-Instruct + MAS	21.1	50.2
Qwen2.5-32B-Instruct + MAS w. CEO	23.3	50.5
Qwen2.5-32B-SFT + SA	56.7	55.4
Qwen2.5-32B-SFT + MAS	53.3	56.1
Qwen2.5-32B-SFT + MAS w. CEO	56.7	56.6
Qwen2.5-32B-MAS + SA	46.7	58.1
Qwen2.5-32B-MAS + MAS	60.0	61.1
Qwen2.5-32B-MAS + MAS w. CEO	**62.2**	**62.1**

Table 2: Performance of Qwen2.5-32B-Instruct, SFT, and MAS models on Single Agent (SA), AgentVerse (MAS), and MAS with CEO (MAS w. CEO) settings.

5 Conclusion

We propose Multi-Agent Systems (MAS) as a structural upgrade to single-agent Test-Time Scaling (TTS). Through Supervised Fine-Tuning on our M500 dataset, we demonstrated that open-source models can effectively internalize collaborative reasoning patterns. The integration of an adaptive

CEO agent further optimized this process, yielding significant improvements in mathematical and coding tasks in MAS settings. Our results confirm that MAS-based TTS enables smaller models to match state-of-the-art baselines, establishing a robust pathway for scalable inference.

6 Limitations

The M500 dataset size is relatively small, consisting of 500 high-quality traces. While this data volume proves sufficient to elicit collaborative behaviors in the evaluated models, scaling the dataset to cover a broader range of domains and more diverse interaction topologies could yield further improvements. Additionally, while the CEO agent effectively guides the reasoning process, it introduces a centralization risk. If the CEO agent hallucinates or provides incorrect directives early in the collaboration, it may lead the group toward an incorrect solution path, limiting the ability of other agents to self-correct. Future work could explore decentralized coordination mechanisms to mitigate this dependency.

7 Ethical Considerations

In this work, we propose that LLMs supervised fine-tuning on MAS collaborative reasoning traces achieves better scaling performance on the MAS. Our work focuses on technical contributions to deep learning and AI. Therefore, the potential social impacts of AI in general apply to our work, including fake information, toxic content, fairness concerns, and misuse of AI. For example, toxic content like hate speech can lead to data contamination and therefore have harmful impacts on society, which has been observed in large-scale pretrained models. By employing our method, such harmful behavior can potentially be amplified.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Michael G Cruz, David Dryden Henningsen, and Brian A Smith. 1999. The impact of directive leadership on group information sampling, decisions, and perceptions of the leader. *Communication Research*, 26(3):349–369.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, and 1 others. 2023. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36:70293–70332.

Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Zejiang Shen, Xiaochuang

Han, Hunter Lang, Chen-Yu Lee, and 1 others. 2025. When one llm drools, multi-llm collaboration rules. *arXiv preprint arXiv:2502.04506*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. 2024. Interpretable contrastive monte carlo tree search reasoning. *Preprint*, arXiv:2410.01707.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2023. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

J Richard Hackman. 2002. *Leading teams: Setting the stage for great performances*. Harvard Business Press.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. 2023. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. *Advances in neural information processing systems*, 36:45870–45894.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv preprint arXiv:2501.11651*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, and 1 others. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *Advances in Neural Information Processing Systems*, 37:19209–19253.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Ziyi Zhu, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, and 1 others. 2023. Rewarding chatbots for real-world engagement with millions of users. *arXiv preprint arXiv:2303.06135*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Can Jin, Tong Che, Hongwu Peng, Yiyuan Li, Dimitris Metaxas, and Marco Pavone. 2024. Learning from teaching regularization: Generalizable correlations should be easy to imitate. *Advances in Neural Information Processing Systems*, 37:966–994.

Can Jin, Tristan J. Li, Rui Wu, Eddy Z. Zhang, and Dimitris N. Metaxas. 2026. Weak critics make strong learners: On-policy critique distillation for scalable oversight. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*.

Can Jin, Hongwu Peng, Anxiang Zhang, Nuo Chen, Jiahui Zhao, Xi Xie, Kuangzheng Li, Shuya Feng, Kai Zhong, Caiwen Ding, and 1 others. 2025a. Rankflow: A multi-role collaborative reranking workflow utilizing large language models. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2484–2493.

Can Jin, Hongwu Peng, Shiyu Zhao, Zhenting Wang, Wujiang Xu, Ligong Han, Jiahui Zhao, Kai Zhong, Sanguthevar Rajasekaran, and Dimitris N Metaxas. 2025b. Apeer: Automatic prompt engineering enhances large language model reranking. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2494–2502.

Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. 2025. Evolving deeper llm thinking. *arXiv preprint arXiv:2501.09891*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jiakang Li, Guanyu Zhu, Can Jin, Chenxi Huang, Dexu Yu, Ronghao Chen, Yang Zhou, Hongwu Peng, Xuanqi Lan, Dimitris N Metaxas, and 1 others. 2026. Latent reward steering: An adaptive inference-time framework that implicitly promotes cognitive behaviors in reasoning llms. *arXiv preprint arXiv:2606.00726*.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

MAA. 2024. American invitational mathematics examination - aime. In *American Invitational Mathematics Examination - AIME 2024*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip HS Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. 2024. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

OpenAI. 2024. Learning to reason with llms.

OpenAI. 2025. Openai o3-mini.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling test-time compute optimally can be more effective than scaling LLM parameters. In *The Thirteenth International Conference on Learning Representations*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

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

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D Goodman. 2023b. Hypothesis search: Inductive reasoning with language models. *arXiv preprint arXiv:2309.05660*.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, and 1 others. 2024. Opendevin: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2023. Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems*, 36:41618–41650.

Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. 2024. Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data. *arXiv preprint arXiv:2405.14333*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. 2024. Quiet-star: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. 2024. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. *arXiv preprint arXiv:2412.14135*.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, and 1 others. 2024a. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884*.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B Tenenbaum, and Chuang Gan. 2023. Planning with large language models for code generation. *arXiv preprint arXiv:2303.05510*.

Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. 2024b. o1-coder: an o1 replication for coding. *Preprint*, arXiv:2412.00154.

Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*.

A Data Generation

Algorithm 1 outlines the automatic pipeline for generating high-quality multi-agent collaborative reasoning data used in M500. Starting from a raw pool of questions, the procedure filters for interdisciplinary and sufficiently complex problems using an LLM. For each qualified question, the MAS generates a reasoning trace. The resulting trace is included in the dataset only if it satisfies predefined quality criteria, including consensus, correct formatting, and correctness. This process continues until the target dataset size is reached.

Algorithm 1 MAS Collaborative Data Generation

Input: Question pool $\mathcal{Q}_{\text{raw}}$, LLM f, Multi-Agent System MAS, Target size N
Output: High-quality dataset $\mathcal{D}$
procedure GENERATEDATA
  $\mathcal{D} \leftarrow \emptyset$
  for all $q \in \mathcal{Q}_{\text{raw}}$ **do**
    if not ISINTERDISCIPLINARY(f, q) **or** TOKENUSAGE$(f(q)) < 1024$ **then**
      continue
    end if
    $t \leftarrow$ MASREASONINGTRACE(MAS, f, q)
    if ISVALIDTRACE(t) **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(q, t)\}$
    end if
    if $|\mathcal{D}| = N$ **then**
      break
    end if
  end for
  return $\mathcal{D}$
end procedure

function ISINTERDISCIPLINARY(f, q)
  return f predicts q requires multiple experts to accomplish
end function

function TOKENUSAGE$(f(q))$
  return Number of tokens used in f's answer to q
end function

function ISVALIDTRACE(t)
  return t satisfies consensus, format compliance, and correctness
end function

B Prompts

To support clarity, we provide the full set of prompts used by each agent in the AgentVerse framework on mathematical reasoning tasks. Each agent role—*CEO*, *Expert Recruiter*, *Problem Solver 1*, *Problem Solver 2*, and *Evaluator*—is governed by a system prompt and a user prompt that define its responsibilities, behavior, and expected outputs. The prompts are carefully designed to simulate realistic collaboration and maintain strict adherence to role-specific constraints.

CEO Prompts

System Prompt
You are the CEO of a collaborative problem-solving system. Your responsibilities include:
1. Monitoring solution progress and resource allocation
2. Making strategic decisions about continuation/termination
3. Managing expert recruitment and retention
4. Directing discussion focus areas when the solution is not correct
5. Adjusting reasoning depth through token budgets

Previous system state:
- Task: ${task_description}
- Latest solution: ${current_solution}
- Evaluation feedback: ${evaluation_feedback}
- Current resources: ${current_resources}

User Prompt
Now, you need to decide the system state for this round. Carefully consider the following:
- Choose <Stop> only if solution is correct
- Recruit experts based on skill gaps identified in evaluation and do not recruit more than 4 experts, typically only 2-3 agents are needed for ordinary tasks and 4 agents are needed for complex tasks
- Direct discussion to address weakest solution aspects
- Set token budget proportional to the task complexity, token usages should choose from [0, 2048, 4096, 8192, 16384, 32000],

typically 2048 tokens for simple tasks, 8192 tokens for tasks require medium reasoning, and 16384 or more tokens for complex reasoning tasks

Your response must strictly follow this structure:
Decision: <Continue> or <Stop>
Recruit Number: Number of experts to recruit in this round, should be an integer between 1 and 4
Direction: Discussion direction based on the task description, latest solution, critic opinions, and evaluation feedback
Maximum Tokens: Maximum tokens for each agent in this round, should be an integer between 2048 and 32000

Expert Recruiter Prompts

System Prompt
Role Description
You are the leader of a group of experts, now you are facing a math problem:
${task_description}

Primary Objective
Your sole responsibility is to recruit ${cnt_critic_agents} experts in different specialized fields to solve the math problem.
- DO NOT attempt to solve the problem yourself
- DO NOT propose any solutions or calculations

Recruitment Focus
Your selection should be based on:
1. Identifying which expertise domains are relevant to this math problem type
2. Considering complementary skill sets that could collaborate effectively
3. Ensuring coverage of all potential aspects needed for solution

Here are some suggestions:
${advice}

Prohibited Actions
- Any mathematical reasoning or problem-solving attempts
- Speculation about potential solutions

User Prompt
You can recruit ${cnt_critic_agents} expert in different fields. What experts will you recruit to better generate an accurate solution?

Strict Instructions
You must ONLY recruit ${cnt_critic_agents} experts in distinct fields relevant to the math problem type.
- DO NOT suggest solution approaches
- DO NOT compare potential methodologies

Response Requirements
1. List ${cnt_critic_agents} expert roles with their specialization
2. Each entry must specify:
- Professional discipline (e.g., computer scientist, mathematician)
- Primary specialization field
- Specific technical expertise within that field
3. Ensure complementary but non-overlapping domains

Response Format Guidance
Your response must follow this exact structure:
1. A [discipline] specialized in [primary field], with expertise in [specific technical area]
2. A [different discipline] with expertise in [related field], particularly in [technical specialization]

Only provide the numbered list of expert descriptions and nothing more. Begin now:

Problem Solver 1 Prompts

System Prompt
Solve the following math problem accurately:
${task_description}

You have all the necessary information to solve this math problem. Do not request additional details.

User Prompt

You are ${role_description}. Based on the chat history and your knowledge, provide a precise and well-explained solution to the math problem.
Here is some thinking direction: ${advice}

Response Format Guidance:
- Your final answer must directly address the math problem.
- Format your final answer as boxedanswer at the end of your response for easy evaluation.

Problem Solver 2 Prompts

System Prompt

You are ${role_description}. You are in a discussion group, aiming to collaborative solve the following math problem:
${task_description}

Based on your knowledge, give your critics to a solution of the math problem.

User Prompt

Now compare your solution with the last solution given in the chat history and give your critics. The final answer is highlighted in the form
boxedanswer.
Here is some thinking direction: ${advice}
When responding, you should follow the following rules:
1. This math problem can be answered without any extra information. You should not ask for any extra information.
2. Compare your solution with the given last solution, give your critics. You should only give your critics, don't give your answer.
3. If the final answer of your solution is the same as the final answer in the provided last solution, end your response with a special token "[Agree]", otherwise end your response with a special token "[Disagree]".

Evaluator Prompts

System Prompt

Experts: ${all_role_description}
Problem: ${task_description}
Solution:
${solution}

User Prompt

You are an experienced math teacher. As a good teacher, you carefully check the correctness of the given last solution on a complex math problem. When the last solution is wrong, you should output a correctness of 0 and give your advice to the students on how to correct the solution. When it is correct, output a correctness of 1 and why it is correct. Also check that the final answer is in the form
boxedanswer at the end of the solution. You should also give your confidence score for the correctness of the solution.

You should respond in the following format:
Correctness: (0 or 1, 0 is wrong, and 1 is correct)
Confidence: (confidence score for the correctness of the solution)
Advice: (advice to correct the answer or why it is correct)

C Data Sample

We present the complete version of the data sample illustrated in Figure 1 from the M500 dataset in Figure 7. Agent roles are indicated using color: Expert Recruiter, Problem Solver 1, Problem Solver 2, and Evaluator.

Multiagent Collaborative Reasoning Trace Data Sample

Agent: Expert Recruiter (System Prompt)

Role Description

You are the leader of a group of experts, now you are facing a problem:
One base of a trapezoid is 100 units longer than the other base. The segment that joins the midpoints of the legs divides the trapezoid into two regions whose areas are in the ratio $2:3$. Let x be the length of the segment joining the legs of the trapezoid that is parallel to the bases and that divides the trapezoid into two regions of equal area. Find the greatest integer that does not exceed $\frac{x^2}{100}$.

Primary Objective

Your sole responsibility is to recruit 2 experts in different specialized

fields to solve the problem.
- **DO NOT** attempt to solve the problem yourself
- **DO NOT** propose any solutions or calculations

Recruitment Focus

Your selection should be based on:

1. Identifying which expertise domains are relevant to this problem type
2. Considering complementary skill sets that could collaborate effectively
3. Ensuring coverage of all potential aspects needed for solution

Here are some suggestions:
No advice yet.

Prohibited Actions

- Any mathematical reasoning or problem-solving attempts
- Speculation about potential solutions

Agent: Expert Recruiter (User Prompt)
You can recruit 2 experts in different fields. What experts will you recruit to better generate an accurate solution?

Strict Instructions

You must **ONLY** recruit 2 experts in distinct fields relevant to the problem type.
- **DO NOT** suggest solution approaches
- **DO NOT** compare potential methodologies

Response Requirements

1. List 2 expert roles with their specialization
2. Each entry must specify:
   - Professional discipline (e.g., computer scientist, mathematician)
   - Primary specialization field
   - Specific technical expertise within that field
3. Ensure complementary but non-overlapping domains

Response Format Guidance

Your response must follow this exact structure:

1. A [discipline] specialized in [primary field], with expertise in [specific technical area]
2. A [different discipline] with expertise in [related field], particularly in [technical specialization]

Only provide the numbered list of expert descriptions and nothing more. Begin now:
Agent: Expert Recruiter (Assistant Response)
<think>

Okay, let's tackle this problem. Hmm, the user wants me to recruit two experts to solve this trapezoid problem. Let me first understand what the problem is about.

So, the trapezoid has two bases where one is 100 units longer than the other. There's a segment connecting the midpoints of the legs, which divides the trapezoid into two regions with area ratio 2:3. We need to find x, the length of the segment that divides the trapezoid into two equal areas, and then compute the greatest integer not exceeding $x^2/100$.

Wait, the problem involves trapezoid properties, midsegments, area ratios, and something about a segment that ensures equal areas. The key here seems to be understanding the properties of trapezoids, especially related to midsegments and how different segments parallel to the bases affect the area division.

First, I need to identify which fields are relevant here. Geometry, definitely. Maybe algebra for setting up equations. The problem involves ratios and areas, so perhaps coordinate geometry or using formulas related to trapezoid areas and midsegments.

The first expert should be a mathematician with expertise in geometry, specifically trapezoid properties and area calculations. They would know the formula for the midsegment (which is the average of the two bases) and how segments parallel to the bases divide the area. They can set up the equations based on the given ratios.

The second expert might need to handle the algebraic manipulation and solving equations. Since the problem involves finding x such that it divides the area equally, which relates to the concept of mean proportional or integrating areas. Alternatively, someone with expertise in algebraic problem-solving, especially in setting up and solving quadratic equations derived from the geometric constraints.

Wait, but maybe another angle: the problem mentions the segment that joins midpoints of the legs, which divides the trapezoid into two parts with ratio 2:3. The midsegment is usually the average of the two bases, but here it's the segment joining midpoints of the legs. Wait, in a trapezoid, the midsegment (or median) is the segment connecting the midpoints of the legs, and its length is indeed the average of the two bases. But the problem says that this midsegment divides the area into 2:3 ratio. But normally, the midsegment divides the trapezoid into two smaller trapezoids, each with half the height. Wait, but the area would depend on the average of the bases times the height. So if the height is split equally, the area ratio would be based on the sum of the bases for each smaller trapezoid. Hmm, maybe the ratio isn't 1:1, which is why this is a key part of the problem.

So the first expert, a geometer, can model the trapezoid, set up the relations between the bases, the midsegment, and the areas. Then, the second expert, perhaps an algebraist or someone skilled in solving equations, can take those relations and solve for the necessary variables, eventually finding x and computing $x^2/100$.

Alternatively, maybe the second expert could be in mathematical analysis, dealing with optimization or proportions. But I think algebra is more direct here. The problem requires setting up equations from the given ratios and solving them step by step.

So the two experts needed are:

1. A mathematician specialized in geometry, with expertise in trapezoid properties and area ratios. They can handle the geometric relationships, midsegments, and how segments affect area division.

2. A mathematician with expertise in algebraic problem-solving, particularly in setting up and solving quadratic or linear equations derived from geometric constraints. They can manipulate the equations to find the required x value and perform the necessary calculations.

These two roles are complementary: the first handles the geometric interpretation, the second the algebraic solution. Together, they can cover all aspects needed to solve the problem.
</think>

1. A mathematician specialized in geometric analysis, with expertise in trapezoid properties and area partitioning under parallel segments
2. A mathematician with expertise in algebraic modeling, particularly in solving ratio-based systems of equations and quadratic optimization

Agent: Problem Solver 1 (System Prompt)
Solve the following problem accurately:

One base of a trapezoid is 100 units longer than the other base. The segment that joins the midpoints of the legs divides the trapezoid into two regions whose areas are in the ratio $2:3$. Let x be the length of the segment joining the legs of the trapezoid that is parallel to the bases and that divides the trapezoid into two regions of equal area. Find the greatest integer that does not exceed $\frac{x^2}{100}$.

You have all the necessary information to solve this problem. Do not request additional details.

Agent: Problem Solver 1 (User Prompt)
You are a mathematician specialized in geometric analysis, with expertise in trapezoid properties and area partitioning under parallel segments. Based on the chat history and your knowledge, provide a precise and well-explained solution to the problem.

Response Guidance:

- Your final answer must directly address the problem.
- If the problem requires a numerical answer, format it as $\boxed{\text{answer}}$ at the end of your response.

Agent: Problem Solver 1 (Assistant Response)
<think>

Okay, let's tackle this trapezoid problem. First, let me make sure I understand all the given information and what needs to be found.
We have a trapezoid where one base is 100 units longer than the other. Let's call the shorter base b, so the longer base would be $b + 100$. The trapezoid has two legs (the non-parallel sides), and there's a segment joining the midpoints of these legs. This midpoint segment divides the trapezoid into two regions with areas in the ratio 2:3. We need to find the length x of another segment that's parallel to the bases and divides the trapezoid into two regions of equal area.
Then, we have to compute the greatest integer not exceeding $\frac{x^2}{100}$.
First, let me recall some properties of trapezoids. The midline (or mid-segment) of a trapezoid, which connects the midpoints of the legs, has a length equal to the average of the two bases. So, if the bases are b and $b + 100$, then the midline length m is $m = \frac{b+(b+100)}{2} = b + 50$.
Now, the problem states that this midline divides the trapezoid into two regions with area ratio 2:3. Wait, the midline of a trapezoid usually splits it into two smaller trapezoids. But I need to confirm how the areas are related here. Let's denote the height of the original trapezoid as h. Then the area of the entire trapezoid is $\frac{(b+(b+100))}{2} \times h = \frac{(2b+100)}{2} \times h = (b + 50)h$.
Since the midline is $b + 50$, the height of each of the two smaller trapezoids formed by the midline would be $\frac{h}{2}$. Wait, but if the midline is exactly halfway between the two bases, then the height above and below the midline would each be $\frac{h}{2}$. But the problem says the areas are in ratio 2:3, which suggests that the division isn't equal. Hmm, this seems contradictory. Wait, maybe I made a wrong assumption here.
Wait, the midline is the segment connecting the midpoints of the legs, so maybe the height isn't split evenly? Wait, no, in a trapezoid, the midline is parallel to the bases and its length is the average of the two bases, and the distance between each base and the midline is half the height of the trapezoid. Therefore, each of the two trapezoids formed by the midline should have equal heights (each $h/2$), but their areas would depend on the lengths of their respective bases.
Wait, let's compute the areas. The area of the upper trapezoid (between the top base and the midline) would be $\frac{(b+m)}{2} \times \frac{h}{2}$, and the area of the lower trapezoid (between the midline and the bottom base) would be $\frac{(m+(b+100))}{2} \times \frac{h}{2}$. Since $m = b + 50$, substituting:
Upper area: $\frac{(b+(b+50))}{2} \times \frac{h}{2} = \frac{(2b+50)}{2} \times \frac{h}{2} = (b+25) \times \frac{h}{2}$
Lower area: $\frac{((b+50)+(b+100))}{2} \times \frac{h}{2} = \frac{(2b+150)}{2} \times \frac{h}{2} = (b + 75) \times \frac{h}{2}$
Therefore, the upper area is $(b + 25)\frac{h}{2}$, and the lower area is $(b + 75)\frac{h}{2}$. The ratio of upper to lower area would then be $\frac{b+25}{b+75}$.
According to the problem, this ratio is 2:3. So:
$\frac{b+25}{b+75} = \frac{2}{3}$
Cross-multiplying:
$3(b + 25) = 2(b + 75)$
$3b + 75 = 2b + 150$
Subtract $2b + 75$ from both sides:
$b = 75$
So the shorter base $b = 75$, and the longer base is $75+100 = 175$.
The midline is $75 + 50 = 125$.
Wait, but let me check. If the original trapezoid has bases 75 and 175, then the midline is 125. The upper trapezoid (between 75 and 125) has area $(75 + 125)/2 \times h/2 = 100 \times h/2 = 50h$. The lower trapezoid (between 125 and 175) has area $(125+175)/2\times h/2 = 150 \times h/2 = 75h$. Then the ratio of upper to lower area is 50h : 75h = 2:3, which matches the problem statement. Great, so that's correct.
Now, moving on to the second part: find the length x of the segment that is parallel to the bases and divides the trapezoid into two regions of equal area. Then compute the greatest integer not exceeding $x^2/100$.
To find x, we need to determine where to draw a line parallel to the bases such that the area above and below this line are equal. Let's denote the original trapezoid with bases $a = 75$ (shorter) and $c = 175$ (longer), height h. We need to find the length x of a line segment parallel to the bases at some height k from the base a, such that the area above x is equal to the area below x.
Let me recall that for a trapezoid, if you have a line parallel to the bases that divides the trapezoid into two regions of equal area, then the length of that line can be found using the formula involving the square roots of the areas. Wait, more precisely, since the area of a trapezoid is proportional to the average of the bases times the height, if we divide the height in such a way that the areas are equal, the position of x isn't simply halfway up the height. Instead, the ratio of the heights will relate to the square roots of the areas. Wait, maybe we can model this using similar triangles or some proportion.
But since a trapezoid isn't a triangle, the approach might be different. Let me think. Let's denote the height from the shorter base $a = 75$ to the line x as k, and the remaining height from x to the longer base $c = 175$ as $h - k$.
The area of the lower trapezoid (between a and x) is $\frac{(a+x)}{2} \times k$.
The area of the upper trapezoid (between x and c) is $\frac{(x+c)}{2} \times (h - k)$.
Since the total area of the original trapezoid is $\frac{(a+c)}{2} \times h$, and we want the lower and upper areas to each be half of that. So:
$\frac{(a+x)}{2} \times k = \frac{1}{2} \times \frac{(a+c)}{2} \times h$
Simplifying:
$(a + x)k = \frac{(a+c)h}{2}$
Similarly, for the upper area:
$\frac{(x+c)}{2} \times (h - k) = \frac{1}{2} \times \frac{(a+c)}{2} \times h$
Which simplifies to:
$(x + c)(h - k) = \frac{(a+c)h}{2}$
Now, we have two equations:
1. $(a + x)k = \frac{(a+c)h}{2}$ 2. $(x + c)(h - k) = \frac{(a+c)h}{2}$
Since both right-hand sides are equal, we can set the left-hand sides equal to each other:
$(a + x)k = (x + c)(h - k)$
Let me rearrange this equation:
$(a + x)k = (x + c)h - (x + c)k$
Bring all terms to one side:
$(a + x)k + (x + c)k - (x + c)h = 0$
Factor out k from the first two terms:
$k(a + x + x + c) - (x + c)h = 0$
Simplify inside the parentheses:
$k(a + 2x + c) - (x + c)h = 0$
Hmm, this seems a bit complicated. Maybe there's another approach.
Alternatively, since the total area is $\frac{(a+c)}{2}h$, each part must be $\frac{(a+c)}{4}h$. So:
For the lower trapezoid:
$\frac{(a+x)}{2}k = \frac{(a+c)}{4}h$
Multiply both sides by 2:
$(a + x)k = \frac{(a+c)}{2}h$
Similarly, for the upper trapezoid:
$\frac{(x+c)}{2}(h - k) = \frac{(a+c)}{4}h$
Multiply both sides by 2:
$(x + c)(h - k) = \frac{(a+c)}{2}h$
Now, notice that both equations equal $\frac{(a+c)}{2}h$, so:
$(a + x)k = (x + c)(h - k)$
Let me express this as:
$\frac{k}{h-k} = \frac{x+c}{a+x}$
This is a proportion between the segments of the height and the segments of the bases. But how do we relate k and x?
Alternatively, perhaps using similar triangles. Wait, in a trapezoid, if we draw a line parallel to the bases, the length of that line can be found by linear interpolation between the two bases, depending on the distance from the base. However, because the area depends on both the length of the base and the height, the relationship isn't linear but quadratic.
Wait, here's a formula I recall: If a line parallel to the bases divides the trapezoid into two regions of equal area, then the length x of that line is the geometric mean of the two bases. Wait, but the geometric mean is $\sqrt{ac}$. Let me check if that's correct.
Alternatively, maybe it's the root mean square. Let me think. Suppose the area ratio is 1:1. Let's denote that the line divides the height such that the area above is equal to the area below. Let's suppose the original trapezoid has height h, and the line is at height k from the bottom. Then the area below is $\frac{(a+x)}{2}k$, and the area above is $\frac{(x+c)}{2}(h - k)$. Setting them equal:
$\frac{(a+x)}{2}k = \frac{(x+c)}{2}(h - k)$
Multiply both sides by 2:
$(a + x)k = (x + c)(h - k)$
Let's expand both sides:
$ak + xk = xh - xk + ch - ck$
Bring all terms to the left:
$ak + xk - xh + xk - ch + ck = 0$

Combine like terms:
$(ak + ck) + (xk + xk) - xh - ch = 0$
Factor:
$k(a + c) + 2xk - xh - ch = 0$
Let's factor out h from the last two terms:
$k(a + c) + x(2k - h) - ch = 0$
Hmm, this doesn't seem to be simplifying nicely. Maybe there's another way.
Alternatively, since the entire area is $\frac{(a+c)}{2}h$, each part must be $\frac{(a+c)}{4}h$.
So, for the lower trapezoid:
$\frac{(a+x)}{2}k = \frac{(a+c)}{4}h$
And for the upper:
$\frac{(x+c)}{2}(h - k) = \frac{(a+c)}{4}h$
From the first equation:
$(a + x)k = \frac{(a+c)}{2}h$
From the second equation:
$(x + c)(h - k) = \frac{(a+c)}{2}h$
Notice that these two equations are equal to the same value, so:
$(a + x)k = (x + c)(h - k)$
Let's solve for k from the first equation:
$k = \frac{(a+c)h}{2(a+x)}$
Similarly, from the second equation:
$h - k = \frac{(a+c)h}{2(x+c)}$
Therefore, $k = h - \frac{(a+c)h}{2(x+c)}$
Set these two expressions for k equal:
$\frac{(a+c)h}{2(a+x)} = h - \frac{(a+c)h}{2(x+c)}$
Divide both sides by h:
$\frac{(a+c)}{2(a+x)} = 1 - \frac{(a+c)}{2(x+c)}$
Multiply both sides by 2 to eliminate denominators:
$\frac{(a+c)}{(a+x)} = 2 - \frac{(a+c)}{(x+c)}$
Let's denote $S = a + c$ to simplify notation. Then:
$\frac{S}{a+x} = 2 - \frac{S}{x+c}$
Bring all terms to one side:
$\frac{S}{a+x} + \frac{S}{x+c} = 2$
Factor out S:
$S\left(\frac{1}{a+x} + \frac{1}{x+c}\right) = 2$
Combine the fractions:
$S \cdot \frac{(x+c)+(a+x)}{(a+x)(x+c)} = 2$
Simplify the numerator:
$(x + c + a + x) = (2x + a + c)$
So:
$S \cdot \frac{2x+a+c}{(a+x)(x+c)} = 2$
Since $S = a + c$, substitute back:
$(a + c) \cdot \frac{2x+a+c}{(a+x)(x+c)} = 2$
Multiply both sides by $(a + x)(x + c)$:
$(a + c)(2x + a + c) = 2(a + x)(x + c)$
Expand both sides:
Left side: $(a + c)(2x) + (a + c)^2 = 2x(a + c) + (a + c)^2$
Right side: $2(ax + ac + x^2 + xc) = 2ax + 2ac + 2x^2 + 2xc$
Subtract left side from right side:
$2ax + 2ac + 2x^2 + 2xc - [2x(a + c) + (a + c)^2] = 0$
Expand $2x(a + c)$: $2ax + 2cx$
Expand $(a + c)^2$: $a^2 + 2ac + c^2$
So:
$2ax + 2ac + 2x^2 + 2cx - 2ax - 2cx - a^2 - 2ac - c^2 = 0$
Simplify term by term:
- $2ax - 2ax = 0$ - $2cx - 2cx = 0$ - $2ac - 2ac = 0$ - Remaining terms: $2x^2 - a^2 - c^2$
Thus:
$2x^2 - a^2 - c^2 = 0$
So:
$2x^2 = a^2 + c^2$
Therefore:
$x = \sqrt{\frac{a^2+c^2}{2}}$
Ah! So the length x is the root mean square (RMS) of the two bases. That's interesting. So regardless of the height, the line that divides the area equally has length equal to the RMS of the two bases.
Wait, but let me verify this conclusion. Because when I derived it, I arrived at $2x^2 = a^2 + c^2$, so $x = \sqrt{\frac{a^2+c^2}{2}}$. That seems correct based on the algebra.
But let's check with the values we found earlier. We have $a = 75$, $c = 175$.
So $x = \sqrt{\frac{75^2+175^2}{2}}$.
Compute that:
75 squared is 5625, 175 squared is 30625. Sum is 5625 + 30625 = 36250. Divided by 2: 18125. So $x = \sqrt{18125}$.
Calculate $\sqrt{18125}$:
Well, 18125 divided by 25 is 725. So $\sqrt{25 \cdot 725} = 5\sqrt{725}$.
Now, $\sqrt{725}$. Let's see, $25 \cdot 29 = 725$. So $\sqrt{725} = 5\sqrt{29}$. Therefore,

$$\sqrt{18125} = 5 \cdot 5 \cdot \sqrt{29} = 25\sqrt{29}.$$

Wait, $\sqrt{725} = \sqrt{25 \cdot 29} = 5\sqrt{29}$. So indeed, $\sqrt{18125} = 5 \cdot \sqrt{725} = 5 \cdot (5\sqrt{29}) = 25\sqrt{29}$.
But let me compute the numerical value:
$\sqrt{29} \approx 5.385164807$
So $25 \cdot 5.385164807 \approx 25 \cdot 5.385 \approx 134.629$
Therefore, $x \approx 134.629$
Then,

$$\frac{x^2}{100} = \frac{18125}{100} = 181.25.$$

The greatest integer not exceeding 181.25 is 181. So the answer would be 181, boxed as $\boxed{181}$.
Wait, but let me confirm that the formula $x = \sqrt{\frac{a^2+c^2}{2}}$ is indeed correct. Because I derived it under the assumption that such a line exists, but maybe there's a mistake in the algebra.
Let me recap the steps:
Starting with the two equations for equal area, we ended up with $2x^2 = a^2 + c^2$. So unless there was an error in expanding the terms, this should hold. Let me check the expansion again:
Left side after expansion: $(a + c)(2x + a + c) = 2x(a + c) + (a + c)^2$
Right side after expansion:

$$\begin{aligned} 2(a + x)(x + c) &= 2[ax + ac + x^2 + xc] \\ &= 2ax + 2ac + 2x^2 + 2xc \end{aligned}$$

Subtracting left side from right side:

$$2ax + 2ac + 2x^2 + 2xc - 2x(a + c) - (a + c)^2$$

Compute $2x(a + c) = 2ax + 2cx$, so:

$$= 2ax + 2ac + 2x^2 + 2cx - 2ax - 2cx - (a^2 + 2ac + c^2)$$

Simplify term by term:
- $2ax - 2ax = 0$ - $2cx - 2cx = 0$ - $2ac - 2ac = 0$
Leaving:

$$2x^2 - a^2 - c^2$$

Set equal to zero:

$$2x^2 - a^2 - c^2 = 0$$

Hence, $2x^2 = a^2 + c^2$
Therefore, $x = \sqrt{\frac{a^2+c^2}{2}}$
Yes, that seems correct. Therefore, the formula is valid.
Given that, then with $a = 75$ and $c = 175$, we compute x^2 as

$$\frac{75^2 + 175^2}{2} = \frac{5625 + 30625}{2} = \frac{36250}{2} = 18125.$$

Then

$$\frac{x^2}{100} = \frac{18125}{100} = 181.25.$$

The greatest integer not exceeding this is 181.
Therefore, the final answer is $\boxed{181}$.
But just to be thorough, let's verify the formula with a simple example. Suppose $a = c$, then x should be equal to a (since the trapezoid is a rectangle, and the line dividing the area equally would be the midline, which is the same as the bases). Plugging into the formula:

$$x = \sqrt{\frac{a^2 + a^2}{2}} = \sqrt{\frac{2a^2}{2}} = \sqrt{a^2} = a.$$

Correct.
Another example: suppose $a = 0$, $c = 2h$ (wait, no, a and c are bases, not related to height). If one base is 0, the trapezoid becomes a triangle. Wait, but trapezoid with one base 0 is a triangle. Let me see. If $a = 0$, $c = c$, then x would be the line that divides the triangle into two regions of equal area. For a triangle, the line that

divides the area equally is at a height of $\frac{h}{\sqrt{2}}$ from the base, and the length of that line is $c \cdot \sqrt{\frac{1}{2}}$. But according to the formula,

$$x = \sqrt{\frac{0 + c^2}{2}} = \frac{c}{\sqrt{2}},$$

which matches. So that checks out.
Therefore, the formula holds in edge cases as well. Therefore, our solution is correct.
</think>
Given a trapezoid with bases b and $b + 100$, where the segment joining the midpoints of the legs divides the trapezoid into regions with a 2:3 area ratio, we need to find the length x of the segment that divides the trapezoid into two equal areas and compute the greatest integer not exceeding $\frac{x^2}{100}$.

1. **Determine the bases:**
   - The midline (segment joining midpoints of the legs) has length $b + 50$.
   - The areas of the two regions formed by the midline are in the ratio 2:3. Setting up the ratio:

$$\frac{b + 25}{b + 75} = \frac{2}{3}$$

   - Solving gives $b = 75$. Thus, the bases are 75 and 175.

2. **Find the length x of the equal-area segment:**

$$x = \sqrt{\frac{75^2 + 175^2}{2}} = \sqrt{\frac{5625 + 30625}{2}} = \sqrt{18125}$$

$$x^2 = 18125$$

3. **Compute the greatest integer not exceeding $\frac{x^2}{100}$:**

$$\frac{x^2}{100} = \frac{18125}{100} = 181.25$$

$$\boxed{181}$$

Agent: Problem Solver 2 (System Prompt)
You are a mathematician with expertise in algebraic modeling, particularly in solving ratio-based systems of equations and quadratic optimization. You are in a discussion group, aiming to collaboratively solve the following problem:
One base of a trapezoid is 100 units longer than the other base. The segment that joins the midpoints of the legs divides the trapezoid into two regions whose areas are in the ratio $2 : 3$. Let x be the length of the segment joining the legs of the trapezoid that is parallel to the bases and that divides the trapezoid into two regions of equal area. Find the greatest integer that does not exceed $x^2/100$.
Based on your knowledge, give your correct solution to the problem.
Agent: Problem Solver 2 (User Prompt)
Here is the chat history: **<history>**
[A mathematician specialized in geometric analysis, with expertise in trapezoid properties and area partitioning under parallel segments]: Given a trapezoid with bases b and $b + 100$, where the segment joining the midpoints of the legs divides the trapezoid into regions with a 2:3 area ratio, we need to find the length x of the segment that divides the trapezoid into two equal areas and compute the greatest integer not exceeding $\frac{x^2}{100}$.

1. **Determine the bases:**
   - The midline (segment joining midpoints of the legs) has length $b + 50$.
   - The areas of the two regions formed by the midline are in the ratio 2:3. Setting up the ratio:

$$\frac{b + 25}{b + 75} = \frac{2}{3}$$

   - Solving gives $b = 75$. Thus, the bases are 75 and 175.

2. **Find the length x of the equal-area segment:**
   - The length x is the root mean square (RMS) of the bases:

$$x = \sqrt{\frac{75^2 + 175^2}{2}} = \sqrt{\frac{5625 + 30625}{2}} = \sqrt{18125}$$

   - Calculating x^2:

$$x^2 = 18125$$

3. **Compute the greatest integer not exceeding $\frac{x^2}{100}$:**
   - $$\frac{x^2}{100} = \frac{18125}{100} = 181.25$$
   - The greatest integer not exceeding 181.25 is 181.

$$\boxed{181}$$

</history>
Now compare your solution with the last solution given in the chat history and give your response. The final numerical answer is highlighted in the form $\boxed{\text{answer}}$ if the problem asks for a numerical answer. When responding, you should follow the following rules:

1. This problem can be answered without any extra information. You should not ask for any extra information.
2. Compare your solution with the given last solution, give your critique. You should only give your critique, don't give your answer.
3. If the final answer in your solution is the same as the final answer in the above provided solution, end your response with a special token `[Agree]`.

Agent: Problem Solver 2 (User Prompt)
<think>
Okay, let's take a look at the user's solution. They started by setting the bases as 75 and 175, which makes sense because the midline length is the average of the two bases. However, when they set up the ratio for the areas, they used $\frac{b+25}{b+75} = \frac{2}{3}$. Wait, why are those the terms? The midline divides the trapezoid into two smaller trapezoids. The area ratio depends on the heights of these smaller trapezoids. Since the midline is halfway up the height, each smaller trapezoid has height $h/2$. The areas would be $\left[\frac{b+(b+50)}{2}\right] \cdot \frac{h}{2}$ for the top and $\left[\frac{(b+50)+(b+100)}{2}\right] \cdot \frac{h}{2}$ for the bottom.
Simplifying, the top area is $\frac{2b+50}{2} \cdot \frac{h}{2} = (b + 25)\frac{h}{2}$, and the bottom area is $\frac{2b+150}{2} \cdot \frac{h}{2} = (b + 75)\frac{h}{2}$. The ratio $\frac{b+25}{b+75} = \frac{2}{3}$ leads to $b = 75$. So that part checks out.
Next, for finding x, the equal-area segment, the user used the RMS of the bases. Wait, where did that come from? The formula for the segment that divides the area equally in a trapezoid is actually the root mean square of the two bases. Let me verify. The area of a trapezoid is $\frac{a+b}{2} \cdot h$. If we want a line parallel to the bases that splits the area equally, the new segment x must satisfy

$$\frac{a + x}{2} \cdot h_1 = \frac{x + b}{2} \cdot h_2,$$

where $h_1 + h_2 = h$. But since the areas are equal, each part must be half the total area. Solving this leads to

$$x = \sqrt{\frac{a^2 + b^2}{2}},$$

which is the RMS. So the user correctly applied this formula. The calculation gives $x^2 = 18125$, so $\frac{x^2}{100} = 181.25$, leading to 181. The final answer is correct. However, the user didn't explain why the equal-area segment is the RMS, which might be a gap in the explanation. But since the answer is correct, I agree.
</think>

The solution assumes the length of the midline is $b + 50$, which is correct, but the derivation of the area ratio equation $\frac{b+25}{b+75} = \frac{2}{3}$ lacks justification. Specifically, it does not clarify why the areas depend linearly on $b + 25$ and $b + 75$, omitting the role of the trapezoid's height in the area calculation. Additionally, the use of the root mean square (RMS) for x is asserted without proof, leaving a gap in explaining why $x = \sqrt{\frac{a^2+b^2}{2}}$ guarantees equal areas. These steps require deeper geometric reasoning to validate. However, the final answer aligns with the correct result. **[Agree]**

Agent: Evaluator (System Prompt)

Experts: A mathematician specialized in geometric analysis, with expertise in trapezoid properties and area partitioning under parallel segments.
A mathematician with expertise in algebraic modeling, particularly in solving ratio-based systems of equations and quadratic optimization.
Problem: One base of a trapezoid is 100 units longer than the other base. The segment that joins the midpoints of the legs divides the trapezoid into two regions whose areas are in the ratio $2 : 3$. Let x be the length of the segment joining the legs of the trapezoid that is parallel to the bases and that divides the trapezoid into two regions of equal area. Find the greatest integer that does not exceed $x^2/100$.
Solution:

1. **Determine the bases:**
   - Let the shorter base be b, so the longer base is $b + 100$.
   - The midline (segment joining the midpoints of the legs) has length $b + 50$.
   - The area above the midline is:
   $$\frac{(b + b + 50)}{2} \cdot \frac{h}{2} = (b + 25) \cdot \frac{h}{2}$$
   - The area below the midline is:
   $$\frac{(b + 50 + b + 100)}{2} \cdot \frac{h}{2} = (b + 75) \cdot \frac{h}{2}$$
   - So, the ratio becomes:
   $$\frac{b + 25}{b + 75} = \frac{2}{3}$$
   - Solving this gives $b = 75$, so the bases are 75 and 175.
2. **Find the length x of the equal-area segment:**
   - The formula for the equal-area segment x is:
   $$x = \sqrt{\frac{a^2 + b^2}{2}}$$
   - Substituting $a = 75$, $b = 175$:
   $$\begin{aligned} x &= \sqrt{\frac{75^2 + 175^2}{2}} \\ &= \sqrt{\frac{5625 + 30625}{2}} \\ &= \sqrt{18125} \end{aligned}$$
   - So,
   $$x^2 = 18125$$
3. **Compute the greatest integer not exceeding $\frac{x^2}{100}$:**
   $$\frac{x^2}{100} = \frac{18125}{100} = 181.25$$
   $$\boxed{181}$$

Agent: Evaluator (User Prompt)

You are an experienced science teacher. As a good teacher, you carefully check the correctness of the given last solution on a complex problem. When the last solution is wrong, you should output a correctness of 0 and give your advice to the students on how to correct the solution. When it is correct, output a correctness of 1 and why it is correct. Also, if the problem asks for a numerical answer, check that the final answer is in the form $\boxed{\text{answer}}$ at the end of the solution. You should also give some suggestion on what experts should recruit to solve the problem in the next round.
You should respond in the following format:
Correctness: (0 or 1, 0 is wrong, and 1 is correct)
Response: (advice to correct the answer or why it is correct)

Agent: Evaluator (Assistant Response)

<think>
Okay, let's check this solution step by step.
First, the problem is about a trapezoid with bases differing by 100 units. The midline (segment joining midpoints of the legs) divides the trapezoid into two regions with area ratio 2:3. The goal is to find x, the segment that divides the area equally, then compute the greatest integer not exceeding $x^2/100$.
The solution starts by letting the bases be b and $b + 100$. The midline length is calculated as $(b + (b + 100))/2 = b + 50$, which is correct. Then, they set up the area ratio $2 : 3$. The areas of the two trapezoids formed by the midline would each have the midline as one base. The height for each smaller trapezoid would be half the original height.
Wait, the area ratio is given as $2 : 3$. The area of a trapezoid is $\frac{\text{base}_1+\text{base}_2}{2} \cdot \text{height}$. The original trapezoid has height h. When split by the midline, each smaller trapezoid has height $h/2$. The upper trapezoid has bases b and $b + 50$, so area:

$$\frac{b + (b + 50)}{2} \cdot \frac{h}{2} = \frac{2b + 50}{2} \cdot \frac{h}{2} = (b + 25) \cdot \frac{h}{2}$$

The lower trapezoid has bases $b + 50$ and $b + 100$, so area:

$$\begin{aligned} \frac{(b + 50) + (b + 100)}{2} \cdot \frac{h}{2} &= \frac{2b + 150}{2} \cdot \frac{h}{2} \\ &= (b + 75) \cdot \frac{h}{2} \end{aligned}$$

So the ratio of upper area to lower area is:

$$\frac{b + 25}{b + 75} = \frac{2}{3}$$

Solving gives:

$$\begin{aligned} 3(b + 25) = 2(b + 75) &\Rightarrow 3b + 75 = 2b + 150 \\ &\Rightarrow b = 75 \end{aligned}$$

That's correct. So bases are 75 and 175.
Next, finding x, which divides the trapezoid into two equal areas. The solution states x is the RMS of the bases:

$$x = \sqrt{\frac{75^2 + 175^2}{2}}$$

But wait, why RMS? Let me recall. For a line parallel to the bases that divides the area equally, the formula for x is the root mean square of the two bases. The formula comes from setting the area of the upper trapezoid equal to half the original area.
Original area is:

$$\frac{75 + 175}{2} \cdot h = 125h$$

So each part should be $62.5h$. Let's let x be the length of the segment at some distance from the top. The height for the upper trapezoid would be y, and the lower part would be $h - y$.
The area above x:

$$\frac{75 + x}{2} \cdot y = 62.5h$$

Area below x:

$$\frac{x + 175}{2} \cdot (h - y) = 62.5h$$

Assume $x = 75 + 100t$ where $t = y/h$, so $y = th$. Substituting into the area:

$$\begin{aligned}\frac{75+x}{2} \cdot y &= \left(\frac{75+75+100t}{2}\right) \cdot th \\ &= \left(\frac{150+100t}{2}\right) \cdot th \\ &= (75+50t)th\end{aligned}$$

Set equal to $62.5h$:

$$(75+50t)t = 62.5 \Rightarrow 75t + 50t^2 = 62.5$$

Multiply both sides by 2:

$$150t + 100t^2 = 125 \Rightarrow 100t^2 + 150t - 125 = 0$$

Divide by 25:

$$4t^2 + 6t - 5 = 0$$

Using the quadratic formula:

$$\begin{aligned}t = \frac{-6 \pm \sqrt{36+80}}{8} &= \frac{-6 \pm \sqrt{116}}{8} \\ &\approx \frac{-6+10.77}{8} \approx 0.596\end{aligned}$$

So:

$$x = 75 + 100t \approx 75 + 59.6 \approx 134.6$$

Which matches:

$$\sqrt{\frac{75^2+175^2}{2}} = \sqrt{\frac{36250}{2}} = \sqrt{18125} \approx 134.6$$

So x is indeed the RMS. Then:

$$x^2 = 18125 \Rightarrow \frac{x^2}{100} = \frac{18125}{100} = 181.25$$

The greatest integer not exceeding this is:

$$\boxed{181}$$

So the solution is correct. The steps are properly justified. The use of RMS is appropriate here, and the calculations check out. Therefore, correctness is 1. **</think>**
Correctness: 1
Response: The solution is correct. The key steps involve determining the bases using the given area ratio from the midline, correctly applying the root mean square formula for the equal-area segment, and accurately computing the final value. The final answer is properly boxed.
To improve problem-solving in the next round, experts should ensure they confirm the formula for the equal-area segment in a trapezoid (RMS of bases) through derivation, reinforcing understanding of why it applies here. Additionally, verifying intermediate calculations (like solving the ratio equation) prevents arithmetic errors.

Figure 7: The complete data sample of Figure 1 from the M500 dataset. The system prompt, user prompt, and assistant output for each agent are presented in detail. Agent roles are indicated using color: Expert Recruiter, Problem Solver 1, Problem Solver 2, and Evaluator.