Collaborative Reasoning: Multi-Agent Small Models for Complex Mathematic Reasoning Tasks

Luning Wang, Larnell Moore, Brandon Zhang, John Kim, Junkuan Liu
{lnwang, larnell, bdwzhang, johnkimm, junkuan}@umich.edu

Abstract

This paper introduces a collaborative, training-free, multi-agent reasoning framework that leverages small language models (SLMs) to solve complex mathematical problems efficiently. While large language models (LLMs) such as GPT-4o demonstrate strong reasoning capabilities, their substantial computational demands, often exceeding 200 billion parameters, make them impractical for many real-world applications. To address this limitation, we propose a lightweight pipeline where multiple diverse SLMs independently attempt a reasoning task, and a larger summarizer model identifies consensus or initiates iterative refinement. In cases of disagreement, each SLM takes on the role of a teacher, critically evaluating anonymized "student" responses (the answers previously generated by the SLMs) and providing justification for its preferred solution. This cycle of "student answer" evaluation continues until convergence is reached or a maximum number of iterations is exceeded. Our method enables enhanced reasoning without additional training or supervision. We evaluate the effectiveness of this framework on challenging benchmarks including GSM8K and AGIEval-SAT-Math. Our code is available at `https://github.com/hjjyhj/CSE545Project`.

1 Introduction

State-of-the-art large language models (LLMs), such as GPT-4o and DeepSeek-R1, are equipped with hundreds of billions of parameters, enabling them to tackle a wide range of complex tasks involving advanced reasoning, including mathematical problem solving and program synthesis. While these models demonstrate remarkable performance, their effectiveness comes at a substantial computational cost. Running or fine-tuning such models demands significant GPU memory, energy consumption, and engineering resources. Furthermore, they often rely on techniques such as Reinforcement Learning from Human Feedback (RLHF) and Supervised Fine-Tuning (SFT), which also require additional training time. As a result, the practical deployment of LLMs is largely restricted to organizations with considerable computational and financial capacity.

To address these limitations, recent research has turned toward improving the reasoning capabilities of smaller models. Small language models (SLMs), typically ranging from 2B to 7B parameters, offer a more accessible and efficient alternative for deployment in real-world scenarios. Techniques such as knowledge distillation and process-level reward modeling (PRM) have shown promise in transferring reasoning abilities from LLMs to SLMs. However, these methods still rely on substantial fine-tuning and curated supervision. Another promising yet underexplored direction lies in leveraging multi-agent collaboration, a paradigm historically applied in distributed systems and decision-making, to improve the reasoning capabilities of SLMs at inference time, without additional training.

In this paper, we focus on solving complex mathematical reasoning tasks, which offer a clear, quantitative benchmark for evaluating model performance. Our proposed framework employs three diverse small language models (SLMs), referred to as candidates, each of which independently generates a reasoning path and final answer. A larger summarizer model then evaluates the outputs to determine whether there is consensus among the candidates. If all models agree, the summarizer outputs the shared answer along with a synthesized reasoning process. In cases of disagreement, the summarizer anonymizes and returns the candidates' responses to them, prompting each model to reflect on its own reasoning and reevaluate the alternative solutions. This iterative process continues until all models converge on a common answer or a predefined maximum number of iterations (MAX_ITER) is reached. The details of the proposed framework are discussed further in Section 2.

2 Proposed method

We propose a multi-agent reasoning pipeline that leverages three small language models (SLMs)-Google Gemma-3 4B It, Meta LLaMA 3.2 3B-Instruct, and Qwen 2.5 3B-Instruct-to collaboratively solve complex reasoning tasks. Given a problem, each SLM independently generates a reasoning path along with its final answer. A summarizer model then aggregates these outputs, extracts the final answers and corresponding reasoning processes, and checks for consensus. If all three models agree, the consensus answer and reasoning path are returned as the final output, and the process terminates.

In cases of disagreement, the summarizer anonymizes and reformats each SLM's response into a standardized prompt, presenting each reasoning path and answer as a "student response." The SLMs are then prompted to assume the role of a teacher, evaluating the anonymized student responses, selecting the one they believe is most reasonable, and providing justification for their choice. If an SLM determines that none of the provided answers are correct, it is encouraged to generate a new answer and reasoning path based on its critique. This iterative process continues until the models reach consensus or a predefined maximum number of iterations is reached. If convergence is not achieved, the summarizer selects the most frequently chosen answer as the final output.

By promoting collaboration and structured peer review among SLMs, combined with enforced Chain-of-Thought reasoning, our framework significantly enhances SLM performance on complex tasks at inference time, without requiring any additional training. The specifics steps of our reasoning pipeline can be found in the Appendix 6

The rationale behind this method is based on the premise that SLMs inherently possess reasoning capabilities, but their ability to generate strong reasoning trajectories is often limited by their default decoding strategies. Prior research suggests that longer reasoning paths tend to yield more accurate solutions, as they allow models to engage in a more structured thought process. By leveraging this idea, our method encourages SLMs to generate detailed reasoning process, increasing the likelihood of generating high-quality Chain of Thought pathway. Additionally, through iterative refinement and cross validation between SLMs, we encourage models to self-correct and collaboratively improve their responses by reconsidering alternative perspectives. The disagreement resolution mechanisms further enhance this effect, enabling the system to refine reasoning outputs dynamically rather than relying on a single model's initial response. By structuring SLMs into a cooperative multi-agent framework, we aim to unlock their latent reasoning potential, demonstrat-

ing that even without fine-tuning or extensive external supervision, smaller models can collectively achieve reasoning performance beyond their individual capabilities.

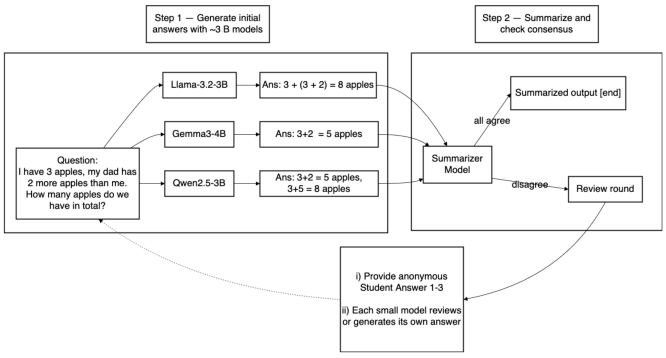


Figure 1: Pipeline diagram

2.1 Candidate and summarizer model

The candidate models were selected based on their performance on benchmark datasets such as GSM8K and MATH-500 under both zero-shot and three-shot settings. The individual accuracy results for each model can be found in Table 6.

We use DeepSeek-V3 as the summarizer model for evaluation due to its strong instruction-following capabilities, which are essential for our setup. Since we rely on regular expressions to extract both the final answer and the reasoning steps, the summarizer must adhere strictly to a predefined output format. We also evaluated the pipeline using Gemma-3 4B as the summarizer and observed comparable accuracy to DeepSeek-V3. In contrast, using Qwen-2.5B resulted in a significant drop in accuracy, primarily because it failed to follow the required output format. These results suggest that as long as the summarizer model exhibits strong instruction-following behavior, the overall accuracy of the pipeline remains consistent.

3 Related work

In recent years, significant efforts have been made to enhance the reasoning abilities of large language models (LLMs) on complex tasks using self-correction and multi-agent strategies. A study by Google DeepMind [4] found that LLMs often struggle to accurately revise their initial responses without external guidance. In some cases, self-correction prompts models to change correct answers into incorrect ones, and overconfidence may cause them to ignore feedback. Similarly, findings from [5] indicate that once an LLM generates an initial solution, it struggles to produce novel reasoning through self-reflection alone. One widely discussed approach for improving chain-of-thought reasoning is the unsupervised self-consistency decoding strategy [8], which samples diverse reasoning paths and selects the most consistent answer via majority voting.

A recent study [9] introduced a Multi-agent Peer-Review Collaboration framework, where each model submits an initial solution, reviews others' reasoning, assigns confidence scores, revises its answer based on peer feedback, and determines its final prediction through a majority vote without

a consensus, significantly outperforming self-correction prompts. The study also found that collaboration is more effective when models have smaller capability gaps or higher diversity, and that stronger models tend to offer more useful feedback. However, the authors noted that the method might be costly.

Another strategy, rStar Math [3], showed that small language models trained with a math policy and process reward model combined with Monte Carlo Tree Search (MCTS) for System 2 style reasoning could rival OpenAI's o1's math performance without distillation, generating step-by-step, verified reasoning paths. Similarly, Corex [7] employed a multi-agent setup where each model was assigned a persona and grouped for iterative discussions. These groups debated factuality and diversity in rationales, reviewed reasoning chains, and ranked the most faithful answers from a pool of candidates.

Several studies have examined the strengths and limitations of multi-agent systems and the use of LLMs as evaluators. One study [6] found that LLM-based judges often display a self-preference bias, favoring outputs from models with similar architectures over those from different models or even humans. This bias can lead to unfair outcomes in collaborative settings and debates, where an agent may prioritize feedback from a similar model, skewing the evaluation process. To mitigate such biases, researchers have explored anonymization techniques. For instance, prior work [1] addressed bias in hiring tasks by removing protected attributes from input data, thereby reducing the influence of sensitive characteristics on model decisions.

Therefore, our work builds such insights and introduces three key novel contributions. Our approach is similar to multi-agent debate systems in that both have multiple agents attempting to answer a question and reach a consensus. However, we introduce the paradigm of using an LLM as a summarizer instead of agents directly debating and considering each other's reasoning. We also anonymize the outputs delivered to each model to combat potential bias in the feedback evaluation step. Also, unlike most multi-agent approaches, which often use the largest and most capable models with extensive training, our approach explicitly intends to investigate the performance of smaller models in collective reasoning without training.

4 Experimental results

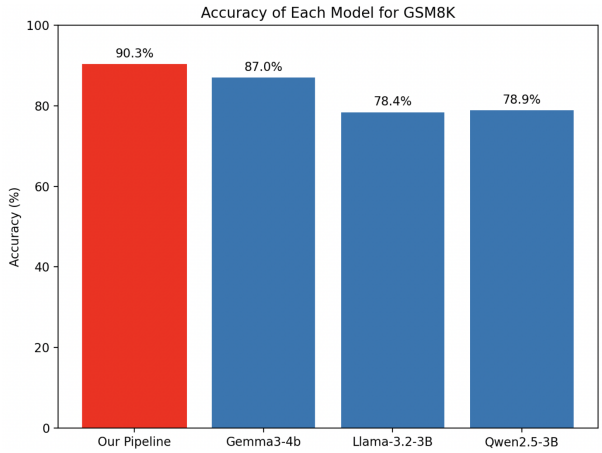
4.1 Qualitative Observation of the Initial Implemented Pipeline

As mentioned in 2, we use Google Gemma-3 4B It, Meta LLaMA 3.2 3B-Instruct, and Qwen 2.5 3B as our small language models. We have evaluated our pipeline's performance using GSM8K and AGIEval-SAT-Math Evaluation. The information about these datasets can be found in the GSM8K and AGIEval subsections of the Appendix.

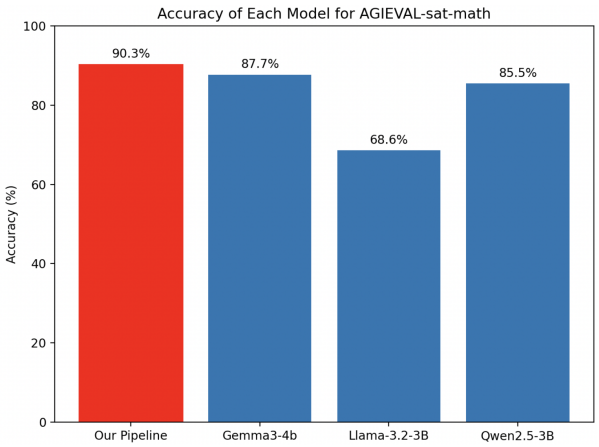
For both GSM8K and AGIEval-SAT-Math, we used DeepSeek-V3 as a summarizer model. For individual model, the questions are only prompted once. As we can see in fig 2, our pipeline achieves the highest accuracy.

4.2 Qualitative Observation of the Finalized Implemented Pipeline

Incorporating feedback from the EECS 545 staff on evaluating our results against a state-of-the-art language model and analyzing how different model combinations affect the pipeline, we developed a second version of the system. This updated version added support for a two-model configuration and included improved parsing methods to enable a fairer comparison with a state-of-the-art baseline.



(a) AGIeval-SAT-Math



(b) GSM8K

Figure 2: Accuracy of our pipeline and individual SLMs across two datasets in our initial pipeline

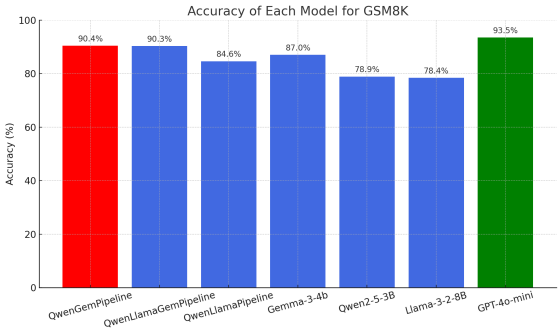
We selected GPT-4o-mini-2024-07-18 as the baseline model for comparison. In our evaluation, accuracy for the individual models was measured using a single pass through the full dataset, with each model prompted only once per question. To assess the pipeline, we compared results using the three following configurations: all three models(QwenLlamaGem), the two best-performing models(QwenGem), and the two lowest-performing models(QwenLlama).

Our results in fig 3 indicate that the pipeline was highly effective in improving accuracy on the AGIEval-SAT Math dataset, even surpassing the performance of GPT-4o-mini. However, the improvement was less significant on the GSM8K dataset. We believe this difference stems from the structural differences between the two datasets. AGIEval-SAT Math consists of multiple-choice questions, which encourage models to use deductive reasoning based on the provided options to make educated guesses or eliminate unlikely answers. Additionally, the limited set of possible answers enables the feedback rounds in the pipeline to be more focused and effective. In contrast, GSM8K requires free-form numerical answers, which provide no such scaffolding and instead demand constructive reasoning. Models must interpret the problem, plan a solution path, and compute an exact answer from start to finish. These findings suggest our framework might be more helpful for tasks that require deductive reasoning given a set of options.

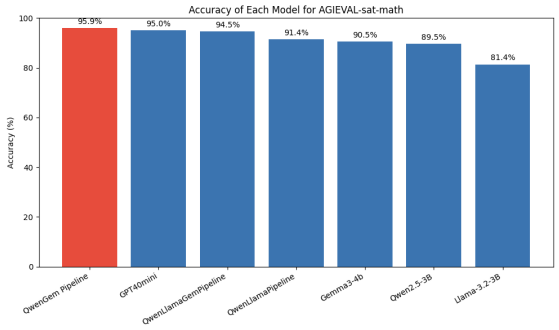
4.3 Behavioral Observations of Models in the Pipeline

The models exhibit various behavioral trends in our experiments that offer further insight into how iterative anonymized feedback and collaboration influenced our results.

Across all three pipeline conditions and models evaluated on the AGIEval-SAT-Math dataset, we observed a consistent pattern during the first iteration, where candidate models must independently generate a solution without external feedback. When a model could not fully comprehend the reasoning behind a multiple-choice option, it often made a best-effort guess, typically selecting the option most closely aligned with its partial reasoning. In cases where the model identified inconsistencies while generating its initial solution, such as when it observed that the answer it concluded did not match any of the available choices, it would sometimes attempt to re-evaluate and correct its reasoning within the same prompt. This process occasionally resulted in the model



(a) GSM8K



(b) AGIEval-SAT-MATH

Figure 3: Accuracy of our pipeline and individual SLMs for GSM8K and AGIEval-SAT-MATH after a revision.

getting stuck within a loop of repeating the same logical errors, ultimately preventing the model from submitting a solution. Furthermore, models that reasoned to a solution often spent the rest of their output verifying their answer by re-calculating the solution or substituting when applicable.

On the GMS8K benchmark, we instead observed that candidate models usually came to one final answer and almost never attempted to verify it. Because the GSM8K problems are simpler than AGI-EVAL-SAT-Math problems, models rarely failed to comprehend the question or submit an answer, except for LLaMA 3.2 3B which often fell into endless loops.

When anonymized feedback was introduced in subsequent iterations for both benchmarks, models commonly adopted a strategy of systematically evaluating each reasoning path. Even when a model reached a confident conclusion, it frequently reviewed the alternative reasoning paths to verify the validity of other options. Additionally, when a model struggled to grasp the underlying logic of a question, it sometimes resorted to deducing the answer through analysis of the anonymized work. In other words, it resorted to the process of elimination and adopted another model's answer without any substantial proof. Some models firmly maintained their answers to specific questions despite being presented with feedback. We observed most false answers typically occurred if the models reached the third iteration, with most answers discovered in the first iteration being correct. In the QwenGem configuration of the pipeline, they frequently agreed on answers, which is expected since these two models had the best reasoning capabilities within the pipeline. At the same time, QwenLlama seemed to have more disagreements, which could have led to its decreased accuracy in our results.

4.4 Secondary Analysis: The Effect Of Anonymized Iterative Reasoning

To evaluate the fairness of comparing our base models to our pipeline, we investigated how applying anonymized iterative reasoning to individual models impacts their performance. In our pipeline, models are allowed up to `MAX_ITERATIONS` to refine their responses unless they converge earlier. However, the baseline in our primary experiments reported accuracy for individual models based on a single iteration through the AGIEval-SAT Math and GSM8K datasets, with no opportunity to receive feedback or self-correct their work. For more challenging examples, iterative refinement may offer an advantage by allowing multiple rounds of self-correction.

This analysis was motivated by feedback from the EECS 545 professor, who questioned whether

a single iteration baseline provided a fair comparison. Drawing inspiration from the Google DeepMind study, we initially hypothesized that iterative refinement would yield only marginal gains and might reduce accuracy in some cases. To verify our baseline, we ran each model with up to `MAX_ITERATIONS=3`, providing an anonymized version of its previous reasoning after each round to guide refinement. Due to time constraints, this evaluation was limited to GPT-4o-mini on AGIEval-SAT-Math, a subset of GSM8K, and Qwen2.5-3B on the AGIEval-SAT-Math dataset.

For Qwen2.5-3B, setting `MAX_ITERATIONS=3` reduced accuracy from 89.5% to 85%, as repeated self-corrections increased divergence from correct solutions. GPT-4o-mini processed 960 GSM8K samples with `MAX_ITERATIONS=3`, achieving an accuracy of 94%, compared to 93% with a single iteration, indicating minimal benefit. On AGIEval-SAT Math, GPT-4o-mini's accuracy improved slightly from 94% to 96%, although this dataset is relatively small compared to GSM8K.

Overall, our findings were consistent with the Google DeepMind study, indicating to us that offering anonymized reasoning paths for an individual model's previous work does not significantly improve results and may even negatively impact them. Thus, our choice of a single iteration with no self-correction mechanism as our baseline remains fair and provides a more accurate interpretation of a model's reasoning capacity compared with our pipeline.

4.5 Effects of Stronger and Weaker Summarizers

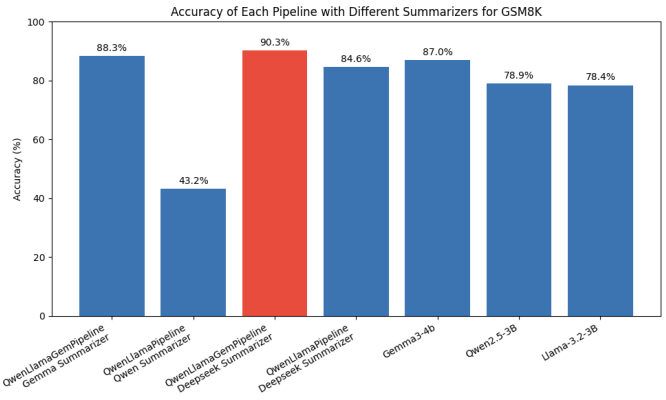


Figure 4: Accuracy of our pipeline with different summarizers and individual SLMs for comparison on GSM8K

We evaluated versions of the pipeline with SLMs as summarizer models on GSM8K to assess how the abilities of the summarizer affect overall pipeline performance. We tested with two different combinations of candidate models, all three SLMs and just LLaMA 3.2 3B and Qwen 2.5 3B. We used the best-performing candidate model (on GSM8k) in each pipeline as the summarizer, Gemma-3 4B for the first and Qwen 2.5 3B for the second. To make the summarizer tasks more manageable for SLMs, the summarizer first summarizes each candidate's answer in a separate context and then determines the consensus/final answer using the summaries.

As shown in 4, the pipelines with DeepSeek-V3 performed better than those with SLMs as the summarizers. The pipeline with Gemma-3 4B as its summarizer and its DeepSeek-V3 counterpart outperformed all the individual SLMs by a non-negligible amount. Still, the pipeline with the Qwen 2.5 3B summarizer performed worse than the individual models.

When Gemma-3 4B is used as a summarizer, it often states that there is no consensus on the candidate answers, even if there is one, causing unnecessary computation of new answers and occasionally allowing the candidate models to propose new, incorrect answers. The Qwen 2.5 3B summarizer also did this but often failed to put the final answer in the correct format. If we count the cases where the Qwen 2.5 3B summarizer put the answer in a slightly different format than expected, the pipeline achieved 80.2% on the benchmark, better than the performance of the individual SLMs in that pipeline.

Together, these results show that the performance of the overall pipeline is dependent on the abilities of the summarizer model. However, the pipeline can still perform better than individual SLMs, even with an SLM used as the summarizer.

5 Limitations and Future Work

The primary limitation of our study arose from time constraints and the limited computational resources available through the Great Lakes Cluster. We did not have enough time to explore the full potential of the pipeline across a wider array of small models, investigate how our pipeline scales with more models, or how our pipeline would perform if we removed the anonymized feedback. Future work could examine how the pipeline performs with a broader selection of small models, larger ensemble size, and an analysis of the effect of the anonymized feedback.

Another limitation was the coarseness of our data analytics, mainly due to time constraints and the effort required to refine the core implementation. Currently, our system displays program output and tracks the number of questions resulting in errors, correct answers, incorrect answers, and overall accuracy for a single run. For future work, we recommend automating the tracking of finer-grained metrics such as which models answered correctly or incorrectly at each iteration, how many iterations each question took to converge, how many questions used majority voting, how responses changed with feedback, how often a model retained its answer despite feedback, and how many iterations were needed for convergence to provide a more comprehensive and quantitative understanding of the pipeline.

An additional limitation is the need for a specific output format. Weaker models like Qwen 2.5 3B may find the correct answer but fail to format it properly, leading to incorrect evaluations. More robust answer processing would improve accuracy.

6 Conclusion

In this project, we propose a novel multi-agent workflow that combines the capabilities of different small models to reach higher accuracy on complex math reasoning tasks. It lets several small models cooperate on a certain problem, and try to reach a consensus after several iterations, under the guidance of a summarizer model. Extensive experiments have shown that our workflow could achieve better accuracy than any of the constituent small models individually, demonstrating the effectiveness of our proposed ideas. We also conduct thorough ablation studies on different combinations of small models and summarizer models to show the characteristics of our workflow. Nevertheless, our work still has some limitations, as we have only experimented to incorporate 2 or 3 small models without testing on a larger scale, and our results are based on two primary-school/SAT level math benchmarks.

Appendix

Pipeline Specifics

Stage 1: Prompt small models and gather outputs
The input prompt containing a question Q is provided to a set of 3 small language models (SLMs) of similar sizes, as we assume different models may possess varying knowledge, leading to more diverse reasoning trajectories. Each model generates an answer and reasoning process, encouraged to give a detailed explanation of how it got the answer. This approach increases the likelihood of capturing high quality Chain of Thought reasoning paths.

Stage 2: Summarizer model checks the answers
Once all reasoning sequences are generated by the candidate SLMs, the summarizer model, larger and more capable than the SLMs, evaluates the answers and corresponding reasoning paths. To ensure that the summarizer acts purely as an aggregator and does not inject its own reasoning into the process, we provide a strict, role-specific prompt. This prompt explicitly instructs the summarizer to operate solely as a passive summarizer without attempting to solve the problem itself.

The prompt is formatted as follows:

```
IMPORTANT: Imagine you are just a summarizer, and you don't have any reasoning
ability.  Make sure to only summarize the answer from the inputs given.
```

Stage 3: Summarizer model checks for consensus
After the summarizer model reviews the answers from the candidate SLMs, if all three models agree, it outputs the consensus answer along with a unified, synthesized reasoning path derived from the candidates' responses.

However, when there is disagreement, the summarizer reformats the candidate responses into a standardized, anonymized format, presenting each as a "student response":

Student 1 answered: `<answer_from_model_1>`
Here is the reasoning process for this answer: `<reasoning_process_for_model_1>` [1em]

Student 2 answered: `<answer_from_model_2>`
Here is the reasoning process for this answer: `<reasoning_process_for_model_2>` [1em]

Student 3 answered: `<answer_from_model_3>`
Here is the reasoning process for this answer: `<reasoning_process_for_model_3>`

The SLMs are then prompted to take on the role of a teacher: they are asked to critically evaluate the student responses and select the answer and reasoning path they believe to be correct. If an SLM determines that none of the provided answers are satisfactory, it is encouraged to generate a new answer and reasoning path that it considers correct. This process fosters self-reflection and peer comparison, contributing to iterative improvement and convergence.

Stage 4: Iterative Consensus Refinement
The newly generated answers from the SLMs are sent back to the summarizer model, and Steps 1, 2, and 3 are repeated until the candidate SLMs reach consensus or the number of iterations reaches the predefined maximum (`MAX_ITER`).

GSM8K

GSM8K [2] is a dataset of high quality linguistically diverse grade school math word problems. The dataset was created to support the task of question answering on basic mathematical problems that require multi-step reasoning. Here we take the test split to perform our evaluation, which has around 1.3K entries.

We refer to an open-source repository [1] to implement the code for evaluation on GSM8K. Due to the limit of running time, we set the maximum number of tokens that the model is allowed to generate (max_new_tokens) to relatively small values, which might influence the results compared with those reported by other papers. Specifically, we have tried on 3 configurations: 1) max_new_tokens=512 with zero-shot; 2) max_new_tokens=512 with 3-shot; 3) max_new_tokens=1024 with zero-shot.

AGIEval

AGIEval [10] is a human-centric benchmark that assesses the general abilities of foundation models regarding human cognition and problem-solving. The benchmark consists of twenty real-world tasks corresponding to standardized admission and qualification exams, such as Math problems derived from the American SAT, LSAT, Gaokao Geography questions, and more. To evaluate the performance of our base models, we ran our tests leveraging the agieval-sat-math task present in the lm-evaluation-harness framework. For these tests, we did not set any token limit. It was tested with default settings under both the few-shot and zero-shot settings. The referenced framework and dataset are noted in our footer [2].

Table 3 depicts the leaderboard of running the AGIEval-SAT-MATH task on our seven models. Qwen2.5-3B-Instruct obtained the best results for the 0-shot and 3-shot settings, showing the strongest mathematical reasoning capabilities by a substantial margin. Deepseek-distill-Qwen-1.5B performed the second best in the zero-shot setting. However, interestingly, the performance saw a decline when applied to the few-shot context. Notably, the three models with the poorest performance in both settings were Zephyr-3B, Llama-3.2-1B-Instruct, and SmolLM2-1.7B-Instruct, which is expected, as they consistently performed poorly in our other benchmarks. Zephyr-3B declined in accuracy with additional examples, while SmolLM2-1.7B-Instruct and Llama-3.2-1B-Instruct had a slight gain in the 3-shot setting. Qwen2.5-1.5B-Instruct performed the third best in the 0-shot and second in the 3-shot setting. Lastly, Llama-3.2-3B-Instruct achieved fourth place in the 0-shot and second in the 3-shot setting. We have also preemptively run the task on other potential models, as observed in Table 2.

Overall, these results indicate that our selected models tend to struggle with complex mathematical reasoning individually. The highest accuracy among them (0.49 in 0-shot, 0.51 in 3-shot) remains significantly below human-level performance. Yet, compared to the results of our AIME and MATH 500 benchmarks, the models performed better under the AGI-MATH-SAT task, with our lowest accuracy on this task being 0.245.

[1] https://github.com/Guangxuan-Xiao/GSM8K-eval

[2] https://github.com/EleutherAI/lm-evaluation-harness/tree/main/lm_eval/tasks/agieval

GSM8K,MATH-500,AIME Table

Table 1: Results for single models on GSM8K and MATH-500 and AIME(1983∼2024)

Setting	Model	Accuracy		
		GSM8K	MATH-500	AIME
MAX_NEW_TOKENS=512, NUM_SHOTS=0	Llama-3.2-1B-Instruct	0.088	-	0.004
	Qwen2.5-1.5B-Instruct	0.318	-	0.004
	SmolLM2-1.7B-Instruct	0.177	-	0.002
	Stablelm-Zephyr-3B	0.080	-	0.008
	Llama-3.2-3B-Instruct	0.741	-	0.060
	Qwen2.5-3B-Instruct	0.732	-	0.012
	Deepseek-R1-Distill-Qwen-1.5B	0.421	-	0.017
MAX_NEW_TOKENS=512, NUM_SHOTS=3	Llama-3.2-1B-Instruct	0.255	0.134	0.008
	Qwen2.5-1.5B-Instruct	0.262	0.068	0.005
	SmolLM2-1.7B-Instruct	0.239	0.172	0.002
	Stablelm-Zephyr-3B	0.497	0.128	0.004
	Llama-3.2-3B-Instruct	0.578	0.290	0.021
	Qwen2.5-3B-Instruct	0.757	0.414	0.016
	Deepseek-R1-Distill-Qwen-1.5B	0.610	0.436	0.023
MAX_NEW_TOKENS=1024, NUM_SHOTS=0	Llama-3.2-1B-Instruct	0.074	-	0.015
	Qwen2.5-1.5B-Instruct	0.321	-	0.008
	SmolLM2-1.7B-Instruct	0.177	-	0.002
	Stablelm-Zephyr-3B	0.080	-	0.008
	Llama-3.2-3B-Instruct	0.737	-	0.084
	Qwen2.5-3B-Instruct	0.713	-	0.050
	Deepseek-R1-Distill-Qwen-1.5B	0.589	-	0.024

AGIEval Table

Table 2: Results on AGIEval-SAT-MATH on Several Models

Setting	Model	Accuracy
NUM_SHOTS=0	Zephyr-3B	0.327
	SmolLM2-1.7B-Instruct	0.291
	Llama-3.2-1B-Instruct	0.245
	Llama-3.2-3B-Instruct	0.359
	Qwen2.5-1.5B-Instruct	0.368
	Qwen2.5-3B-Instruct	0.49
	Deepseek-R1-Distill-Qwen-1.5B	0.3818
	Llama-3.1-8B-Instruct	0.359
	Mistral-7B-Instruct	0.345
	Qwen2.5-7B-Instruct	0.57
	Qwen2.5-Math-7B	0.636
NUM_SHOTS=3	Zephyr-3B	0.2772
	SmolLM2-1.7B-Instruct	0.2954
	Llama-3.2-1B-Instruct	0.28
	Llama-3.2-3B-Instruct	0.35
	Qwen2.5-1.5B-Instruct	0.44
	Qwen2.5-3B-Instruct	0.51
	Deepseek-R1-Distill-Qwen-1.5B	0.3227
	Llama-3.1-8B-Instruct	0.4
	Mistral-7B-Instruct	0.38
	Qwen2.5-7B-Instruct	0.59
	Qwen2.5-Math-7B	0.627

AGIEval Table of our seven selected models

Table 3: Results on AGIEval-SAT-MATH on our 7 models

Setting	Model	Accuracy
NUM_SHOTS=0	Qwen2.5-3B-Instruct	0.49
	Deepseek-R1-Distill-Qwen-1.5B	0.3818
	Qwen2.5-1.5B-Instruct	0.368
	Llama-3.2-3B-Instruct	0.359
	Zephyr-3B	0.327
	SmolLM2-1.7B-Instruct	0.291
	Llama-3.2-1B-Instruct	0.245
NUM_SHOTS=3	Qwen2.5-3B-Instruct	0.51
	Qwen2.5-1.5B-Instruct	0.44
	Llama-3.2-3B-Instruct	0.35
	Deepseek-R1-Distill-Qwen-1.5B	0.3227
	SmolLM2-1.7B-Instruct	0.2954
	Llama-3.2-1B-Instruct	0.28
	Zephyr-3B	0.2772

References

[1] Django Beatty, Kritsada Masanthia, Teepakorn Kaphol, and Niphan Sethi. Revealing hidden bias in ai: Lessons from large language models. *arXiv preprint arXiv:2410.16927*, 2024.

[2] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[3] Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

[4] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

[5] Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2310.03085*, 2023.

[6] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[7] Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv preprint arXiv:2310.00280*, 2023.

[8] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[9] Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. Towards reasoning in large language models via multi-agent peer review collaboration, 2023.

[10] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023.