SLM-Math: Empowering Small Language Models for Complex Mathematical Problem Solving

COMS 4705 Project Proposal

Keywords: *Small Language Models, Mathematical Reasoning, Reinforcement Learning, Agentic AI Systems, Chain-of-Thought*

Roger Wang
Department of Computer Science
Columbia University
lw3240@columbia.edu

Jinzi Luo
Department of Computer Science
Columbia University
jl7199@columbia.edu

Yunchen Yuan
Department of Computer Science
Columbia University
yy3610@columbia.edu

1 Key Information to include

Title:
SLM-Math: Empowering Small Language Models for Complex Mathematical Problem Solving

Keywords:
Small Language Models, Mathematical Reasoning, Reinforcement Learning, Agentic AI Systems, Chain-of-Thought

Team Member Names:

- Roger Wang (lw3240@columbia.edu)
- Jinzi Luo (jl7199@columbia.edu)
- Yunchen Yuan (yy3610@columbia.edu)

External Collaborators: None.
Mentor: No specific mentor requested; open to assignment based on NLP reasoning expertise.
Sharing Project: Exclusive to COMS 4705.

2 AI-Aided Literature Review and Critique

Chosen LLM & Responses. We used ChatGPT-5 and AI2 ScholarQA. For ChatGPT: `https://chatgpt.com/share/69016dcc-f668-8003-ae23-33aba0fb04cd`. For AI2 ScholarQA: See Appendix 3 for the full response.

Question. We asked: *"What modeling or training techniques enable small language models to perform complex mathematical reasoning despite limited parameter counts?"* This question explores how smaller models (under 7B parameters) can achieve reasoning depth similar to much larger LLMs, despite lower compute capacity. While models like GPT-4 demonstrate strong reasoning, they are costly and opaque. Understanding which modeling and training methods—such as reasoning fine-tuning or reward-based self-improvement—allow smaller systems to perform comparably is both scientifically and practically important for efficient, interpretable NLP.

Columbia COMS4705: Natural Language Processing, Fall 2025. Adapted from Stanford CS224N Natural Language Processing with Deep Learning.

Critique of Literature Review. ChatGPT highlights key methods like reasoning distillation from larger models, chain-of-thought, agent-style tool use and etc. It also explains why these help small models with limited capacity. Howver, It doesn't give empirical evidence and doesn't analyze how reliable they are. ScholarQA directly targets my question about math reasoning in small ($\leq$ 7B) models, explains why each paper is relevant, and shows how mature the area is. But it doesn't summarize what the papers actually do, how well the methods work, or discuss key approaches like agent-style reasoning, tool use, or self-verification.

We found that the ScholarQA response is more helpful for the literature-review task, because it directly targets our question about math reasoning in 7B models, explains exactly how it's filtering papers, and reports how many strong matches it found, which helps us judge how mature the area is.

Chosen Article. We selected rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking [1]. The paper investigates how small language models (1.5B parameters) can perform complex mathematical reasoning without relying on large-model supervision. It introduces a self-evolution framework that combines Monte Carlo Tree Search (MCTS), a Process Preference Model (PPM) for ranking reasoning steps, and iterative co-training between policy and reward models. This work directly addresses my question, showing that strong reasoning can emerge from structured search and feedback mechanisms rather than model scale. Compared with prior distillation-based approaches such as MetaMath and NuminaMath, rStar-Math contributes a teacher-free, self-improving method in which the model generates and refines its own reasoning traces. The article was highly relevant and was appropriately discussed in the ScholarQA response but not in ChatGPT's output, underscoring ScholarQA's stronger research grounding.

Reflection. From this review, We learned that strong reasoning in small LLMs can emerge from training strategies rather than model size. The rStar-Math paper was especially interesting for its use of self-evolution and reward-based feedback, showing that models can refine their reasoning autonomously without teacher supervision. We were surprised to see that a 7B-parameter model could reach GPT-4 accuracy through deeper test-time reasoning instead of scale. The answers largely aligned with my expectations but revealed how search and feedback mechanisms can act as an alternative to data distillation. Remaining open questions include how such self-evolving frameworks generalize beyond math—particularly to logical proofs or scientific reasoning domains. These insights motivate our project to explore lightweight reasoning architectures that emphasize interpretability and iterative improvement over parameter growth.

3 Project description

Goal. Our scientific goal is to determine whether reinforcement learning (RL) combined with multi-agent collaboration can substantially enhance the reasoning depth of small language models (8B). Specifically, we ask: *Can Qwen3-1.7B and Qwen3-8B achieve competitive mathematical reasoning through multi-agent RL optimized with both process and outcome rewards?*

It is scientifically interesting as it explores whether emergent collaboration among small, specialized agents can approximate the self-evolutionary learning behaviors of much larger models.

The project is challenging because multi-agent RL introduces complex dynamics—coordination, credit assignment, and reward design—that are not yet well understood in reasoning contexts.

However, it is also likely to succeed, given that our framework builds upon proven foundations from rStar-Math [1] (teacher-free RL reasoning) and MathChat [2] (agentic dialogue reasoning), which together demonstrate complementary strengths.

As secondary goals, we will:

- Explore extensions to multi-agent RL algorithms (e.g., adaptive reward shaping or joint policy optimization) to improve collaboration stability.
- Analyze qualitative aspects such as interpretability, error correction, and reasoning trace coherence to better understand emergent collaboration.

These stretch goals will be pursued if core training and evaluation milestones are achieved ahead of schedule.

Task. The target task is stepwise mathematical problem solving. Given a natural-language math problem x, the model must produce a reasoning trace $(r_1, r_2, \ldots, r_T)$ and a final answer y. For example:

> **Input:** "If a triangle has sides 3, 4, and 5, what is its area?"
> **Output:** "Step 1: Recognize right triangle. Step 2: Compute area $= \frac{3\times4}{2} = 6$.
> Final answer: 6."

This output format supports both automatic and process-level evaluation.

Data. We employ three public datasets of increasing difficulty and scale:

- **GSM8K** [3]: 8,500 grade-school math word problems with step-by-step rationales; used for warm-up training, prompt tuning, and early evaluation.
- **MATH** [4]: 12,500 competition-style problems across algebra, geometry, calculus, and number theory; serves as our main benchmark for complex reasoning.
- **AIME-2024** [5]: 30 problems per annual set (roughly 300 problems in total), representing high-difficulty short-answer questions for advanced evaluation and generalization.

Preprocessing includes normalization into structured `{problem, reasoning, answer}` JSON records, tokenization with the Qwen vocabulary, and filtering for text-only problems to ensure compatibility with small models. If RL exhibits cold-start instability, we will bootstrap with teacher-generated reasoning traces for supervised fine-tuning following [6].

Methods. Our approach follows a three-stage incremental development process aimed at systematically enhancing mathematical reasoning in small language models (SLMs) while maintaining interpretability and efficiency. Each stage builds upon the previous one, with continuous evaluation and refinement.

1. **Stage 1 – Prompt Engineering for Base Model Evaluation.** We first evaluate the baseline reasoning performance of Qwen3-1.7B and Qwen3-8B via prompt engineering (e.g., Chain-of-Thought, Program-of-Thought, least-to-most) to obtain training-free baselines and expose systematic failure modes [7]. No parameters are updated in this phase.
2. **Stage 2 – Building and Optimizing the Agentic Workflow (Training-Free).** We construct a *multi-agent reasoning framework* inspired by MathChat [2] with specialized roles—*Solver*, *Checker*, *Reflector*. We design the dialogue protocol, run training-free experiments, and perform detailed error analysis (trace alignment, inter-agent disagreement, self-correction). We iteratively optimize prompts, role responsibilities, and interaction rules to maximize robustness before learning.
3. **Stage 3 – Reinforcement Learning with Result and Process Supervision.** After stabilizing the workflow, we introduce RL to optimize the agentic system jointly. Guided by rStar-Math's self-evolution insight [1] and multi-agent RL principles (e.g., MAPoRL) [8], we employ a dual reward:
$$R = \alpha R_{result} + (1 - \alpha) R_{process},$$
where R_{result} measures final-answer correctness and $R_{process}$ measures step-wise validity and inter-agent consistency. To mitigate cold-start/sparse-reward issues, we optionally warm up with supervised fine-tuning (SFT) using teacher traces and then adaptively blend SFT and RL as in [6]. This hybridization aligns exploration (RL) with stable supervision (SFT).

Original contribution and motivation. Our main contribution is the integration of an *agentic multi-agent system* with *reinforcement learning* to enable collaborative machematical reasoning in small language models. Unlike prior work that treats multi-agent prompting or RL independently, our method jointly trains interacting agents to learn *how to collaborate*—when to propose, verify, and reflect—within a shared reasoning environment. We further design a robust RL training framework that supports stable multi-agent coordination through reward shaping and process-level feedback, allowing agents to solve complex, multi-step math problems more effectively. This approach is expected to yield higher reasoning accuracy, faster convergence, and improved efficiency across GSM8K [3], MATH [4], and AIME-2024 [5].

Baselines. We evaluate our proposed approach against two representative systems that together define the current state of small-model mathematical reasoning:

- **rStar-Math** [1]: a teacher-free reinforcement learning framework that achieves strong reasoning in small models. We will reproduce its key experiments using the official open-source implementation (`https://github.com/microsoft/rStar`) to validate reported results and ensure a fair comparison. This baseline represents the best-performing single-agent RL paradigm for small models.
- **MathChat** [2]: a multi-agent conversational solver based purely on prompting, without reinforcement learning. We will reference both its public codebase (`https://github.com/Zhenwen-NLP/MathChat`) and published results for comparison. This system exemplifies the strongest training-free agentic reasoning framework.

These baselines are selected because they represent the two dominant directions most comparable to our work: (1) single-agent RL for self-improving reasoning, and (2) multi-agent prompting for collaborative reasoning. Comparing against them allows us to measure the added benefit of integrating RL into an agentic framework. These span both RL-based and dialogue-based paradigms, providing meaningful contrast. rStar-Math (7B) reaches $\sim$90% on GSM8K and 70% on MATH; MathChat (GPT-4) reaches $\sim$45% on level-5 MATH problems. Specifically, our hybrid system aims to combine rStar-Math's learning efficiency with MathChat's collaborative verification, providing insight into whether explicit coordination among agents can outperform isolated optimization or static prompt interaction. We expect this comparison to clarify trade-offs between reasoning robustness, sample efficiency, and computational cost in small language models.

Evaluation. We evaluate our model's reasoning ability primarily through quantitative accuracy and qualitative behavioral analysis.

- **Pass@1 Accuracy:** the proportion of problems for which the model's first predicted answer exactly matches the correct solution. This serves as the main quantitative measure of reasoning performance across datasets.

We target $\geq$80% Pass@1 on GSM8K and $\geq$50% on MATH, which would indicate strong small-model reasoning performance. In addition, we perform **qualitative analysis** of reasoning traces to assess inter-agent disagreement rates, error correction frequency, and the clarity of intermediate reasoning steps—providing insights into how agent collaboration contributes to improved interpretability and stability in problem solving.

Justification and Team Roles. The project fulfills the final-project expectations by integrating model implementation, RL optimization, and multi-agent reasoning analysis within a well-scoped, semester-length framework.

Yunchen Yuan will lead the reinforcement learning design and implementation, including reward modeling, training pipeline development, and performance evaluation. **Jinzi Luo** will focus on constructing the multi-agent orchestration system, defining agent roles, communication strategies, and interaction logic. **Roger Wang** will manage dataset preprocessing, baseline reproduction (rStar-Math and MathChat), and final reporting, ensuring reproducibility and documentation consistency.

This division of labor balances workloads across modeling, systems, and analysis components. We believe the project is feasible within the semester because:

- The required datasets (GSM8K, MATH, AIME) and baseline implementations are all publicly available, including rStar-Math and MathChat, which significantly lowers the engineering overhead.
- Open-source Qwen3 models (1.7B and 8B) are lightweight and can be fine-tuned efficiently on accessible GPU resources.
- Early studies on multi-agent reinforcement learning, such as MAPoRL [8], demonstrate the technical viability of agent-based RL training, providing conceptual guidance for our implementation.

This structured workflow ensures consistent deliverables and meaningful evaluation results by the end of the semester, while remaining grounded in reproducible baselines and supported by prior successful multi-agent RL research.

References

[1] Xinyu Guan, L. 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.org*, 2025.

[2] Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. Mathchat: Converse to tackle challenging math problems with llm agents. *arXiv preprint arXiv:2306.01337*, 2023.

[3] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[4] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[5] Maxwell Jia. AIME_2024: A dataset of 2024 American Invitational Mathematics Examination problems with solutions and answers. `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024`, 2024. Dataset at Hugging Face, tag: explanation-generation, license: MIT.

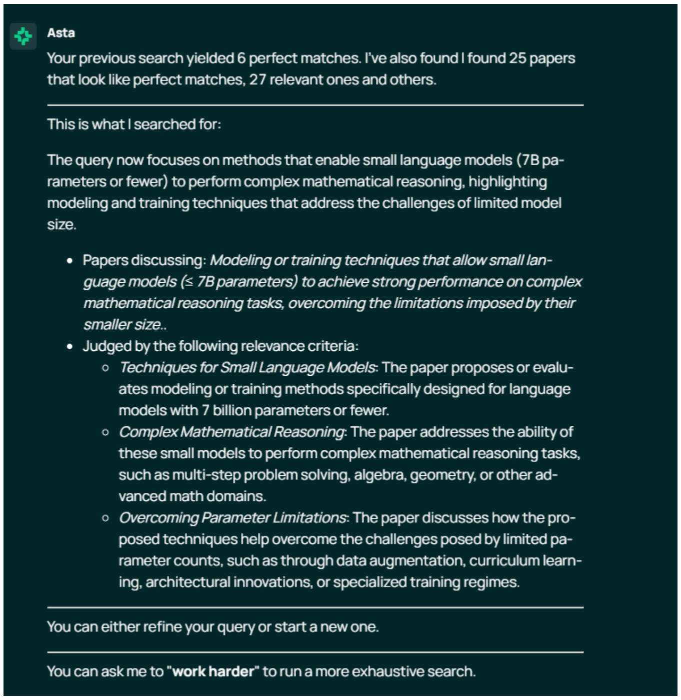
[6] Jack Chen, Fazhong Liu, Naruto Liu, Yuhan Luo, Erqu Qin, Harry Zheng, Tian Dong, Haojin Zhu, Yan Meng, and Xiao Wang. Step-wise adaptive integration of supervised fine-tuning and reinforcement learning for task-specific llms. *arXiv.org*, 2025.

[7] Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning*, pages 10421–10430. PMLR, 2023.

[8] Chanwoo Park, Seungju Han, Xingzhi Guo, A. Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

Appendix: ScholarQA-Cited Excerpt for Small-Model Math Reasoning

The following excerpt from *rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking* was referenced in response to the question: *"What modeling or training techniques enable small language models (7B parameters) to achieve strong performance on complex mathematical reasoning tasks, overcoming the limitations imposed by their smaller size?"*



Asta

Your previous search yielded 6 perfect matches. I've also found I found 25 papers that look like perfect matches, 27 relevant ones and others.

This is what I searched for:

The query now focuses on methods that enable small language models (7B parameters or fewer) to perform complex mathematical reasoning, highlighting modeling and training techniques that address the challenges of limited model size.

- Papers discussing: *Modeling or training techniques that allow small language models (≤ 7B parameters) to achieve strong performance on complex mathematical reasoning tasks, overcoming the limitations imposed by their smaller size..*
- Judged by the following relevance criteria:
    - *Techniques for Small Language Models*: The paper proposes or evaluates modeling or training methods specifically designed for language models with 7 billion parameters or fewer.
    - *Complex Mathematical Reasoning*: The paper addresses the ability of these small models to perform complex mathematical reasoning tasks, such as multi-step problem solving, algebra, geometry, or other advanced math domains.
    - *Overcoming Parameter Limitations*: The paper discusses how the proposed techniques help overcome the challenges posed by limited parameter counts, such as through data augmentation, curriculum learning, architectural innovations, or specialized training regimes.

You can either refine your query or start a new one.

You can ask me to "**work harder**" to run a more exhaustive search.

Figure 1: Response from ScholarQA.