TriMER: Balancing Efficiency and Accuracy in Mathematical Reasoning through a Three-Stage LLM Pipeline

Anonymous ACL submission

Abstract

Despite remarkable advances in Large Language Models (LLMs), mathematical reasoning remains a critical frontier where models struggle with accuracy, reliability, and computational efficiency, particularly for competition-level problems. Current approaches face fundamental limitations: distillation methods alone fail to capture reasoning depth, reinforcement learning techniques demand prohibitive computational resources, and ensemble methods multiply inference costs.We introduce a novel three-stage framework, TriMER (**Tri**ple-stage **M**athematical **E**fficient **R**easoning), that synergistically combines reasoning capability distillation, Group Relative Policy Optimization (GRPO) with zero KL penalty, and multi-agent Preference Reward Model (PRM) reranking to address both reasoning quality and computational efficiency. Leveraging our curated dataset of 387K high-difficulty mathematical problems, we achieve state-of-the-art performance of 76.7% accuracy on the challenging AIME24 benchmark, surpassing Qwen-R1-Distilled-32B (73.3%) and Qwen2.5-Math-72B (30.0%) while using only 5048 tokens per problem—a 6.3× reduction in computational requirements. Our multi-agent framework further improves performance to 79.9% accuracy, demonstrating robustness through solution diversity. Extensive ablation studies confirm the essential contribution of each component, with significant gains from our memory-optimized GRPO implementation `https://anonymous.4open.science/r/math_reasoner-362E/`[1]. By effectively resolving the efficiency-accuracy trade-off that has hindered practical deployment of mathematical reasoning systems, our approach establishes a new paradigm for developing LLMs that can tackle complex mathematical challenges while remaining computationally accessible for real-world applications.

[1]To be released upon publication.

1 Efficient Multi-Stage Optimization for Advanced Mathematical Reasoning in LLMs

Mathematical reasoning remains one of the most challenging frontiers for large language models (LLMs), demanding capabilities fundamentally different from general language understanding. While LLMs have made remarkable progress across many domains, solving complex mathematical problems requires maintaining symbolic consistency, executing multi-step logical reasoning, and applying domain-specific knowledge—capabilities that traditional supervised fine-tuning approaches often fail to develop fully.

Three critical challenges continue to limit the practical application of LLMs for advanced mathematical reasoning:

- **Reasoning depth**: Models struggle to maintain coherent reasoning chains across multiple interdependent steps required for competition-level problems.
- **Computational efficiency**: State-of-the-art performance typically demands excessive token generation, making deployment impractical.
- **Solution verification**: Ensuring answer correctness often requires additional computation that compounds resource requirements.

Current approaches to mathematical reasoning in LLMs have made progress but face significant limitations. Knowledge distillation approaches transfer capabilities from larger teacher models but often sacrifice reasoning depth; reinforcement learning techniques improve reasoning quality but demand prohibitive computational resources; and ensemble methods enhance accuracy by generating multiple solutions but multiply inference costs, making them impractical for real-world applications.

We introduce a novel three-stage framework that synergistically combines the strengths of these approaches while addressing their limitations:

1. **Capability distillation** from DeepSeek-R1 to Qwen2.5 models establishes a strong mathematical reasoning foundation.

2. **Memory-optimized Group Relative Policy Optimization (GRPO)** with zero KL penalty enables efficient scaling to 32B parameter models.

3. **Multi-agent Preference Reward Model (PRM) reranking** leverages solution diversity to significantly improve performance.

Our approach achieves state-of-the-art 76.7% accuracy on the challenging AIME'24 benchmark while using just 5048 tokens per problem—a 6.3× reduction compared to baseline approaches. The multi-agent framework further improves performance to 79.9% accuracy, demonstrating the effectiveness of solution diversity.

This work resolves a critical tension between reasoning quality and computational efficiency in mathematical problem-solving, establishing a new paradigm for developing LLMs that excel at complex mathematical challenges while remaining computationally accessible for practical applications.

2 Related Work

Our work builds on research enhancing mathematical reasoning in LLMs across four key areas: prompt engineering, distillation, reinforcement learning, and verification techniques.

2.1 Prompt Engineering for Mathematical Reasoning

Chain-of-Thought prompting [Wei et al., 2022] demonstrated that showing reasoning steps significantly improves mathematical problem-solving, leading to variants like Zero-shot CoT [Kojima et al., 2022], Self-consistency [Wang et al., 2023b], and Tree-of-Thought [Yao et al., 2023]. While effective, these approaches require extensive token usage, and efficiency-focused strategies like Equation-of-Thought [Zhang et al., 2023] and Chain-of-Draft [Xu et al., 2025] optimize inference patterns without addressing underlying model capabilities.

2.2 Distillation for Mathematical Reasoning

Knowledge distillation transfers reasoning capabilities from larger to smaller models, with DeepSeek-R1 [AI, 2025] demonstrating effective transfer from RL-trained teachers to student models. Our work extends this approach with specialized techniques for mathematical reasoning, using adaptive temperature scheduling and weighted masking to enhance reasoning transfer.

2.3 Reinforcement Learning for Reasoning

MATH-Shepherd [Wang et al., 2023a] and DeepSeekMath [Shao et al., 2024] employed variants of Proximal Policy Optimization [Schulman et al., 2017] for mathematical reasoning, while Group Relative Policy Optimization (GRPO) [Shao et al., 2024] eliminated the separate critic network. We build on GRPO with a novel zero KL penalty formulation that substantially reduces memory requirements while improving mathematical reasoning performance.

2.4 Verification and Multi-agent Approaches

Let's Verify Step by Step [Lightman et al., 2023] demonstrated the value of checking intermediate reasoning steps, while Collaborative Mathematics [Wu et al., 2023] showed benefits of combining multiple reasoning agents. Our multi-agent PRM framework integrates these approaches into a unified system that maximizes reasoning quality while minimizing computational costs.

2.5 Efficiency in Mathematical Reasoning

Recent work like the Agentica Project [Luo et al., 2025] has explored efficiency improvements through specialized training and scaling techniques. Our approach achieves state-of-the-art performance with substantially reduced computational requirements through the synergistic combination of distillation, memory-optimized GRPO, and selective multi-agent verification.

3 Dataset Creation and Curation

We developed a carefully curated dataset of 387K competition-level mathematical problems to support effective training across diverse problem types and difficulty levels.

3.1 Dataset Composition

Our dataset synthesizes three complementary sources: NuminaMath 1.5 [Li et al., 2024] provided competition-level problems with varied difficulty; OpenR1-Math-220K contributed multiple reasoning traces (55% of our final dataset); and Bespoke-Stratos-17K (7.5%) offered meticulously crafted step-by-step solutions for challenging problems. We further enriched this foundation with 150K pipeline-generated solutions (37.5%) specifically designed to address coverage gaps, ensuring comprehensive representation across algebra, geometry, number theory, and combinatorics.

3.2 Quality Assurance

Data quality was ensured through a multi-stage verification pipeline that included automatic deduplication, domain-specific filtering, and solution validity checking. We employed OpenAI-4o-mini as an independent judge to evaluate solution quality, resulting in approximately 15% of initially generated solutions being rejected. This rigorous quality control process maintained high standards while preserving mathematical diversity, creating a robust foundation for training models capable of advanced reasoning across problem types.

Our solution generation pipeline evolved from an initial multi-agent approach requiring approximately 10 minutes per batch to an optimized system that reduced processing time by 70% through contextual leveraging of existing solutions, allowing efficient scaling to our final dataset size while maintaining solution quality. Our curation process prioritized olympiad-level problems requiring sophisticated multi-step reasoning while maintaining balanced representation across difficulty levels to support robust model training.

4 Reasoning Capability Distillation

We developed a specialized distillation framework to transfer mathematical reasoning capabilities from a capable teacher model to a more efficient student architecture, focusing on preserving step-by-step deduction patterns.

Our approach uses DeepSeek-R1 [AI, 2025] as the teacher model and Qwen2.5-32B as the student, with a hybrid loss function balancing reasoning fidelity and answer accuracy:

$$\mathcal{L} = \alpha\mathcal{L}_{KL} + (1-\alpha)\mathcal{L}_{CE} \tag{1}$$

The KL divergence component captures teacher reasoning patterns through temperature-controlled softened logits, while the cross-entropy loss ensures alignment with ground truth solutions:

$$\mathcal{L}_{KL} = T^2 \cdot KL\left(\text{softmax}\left(\frac{z_t}{T}\right) \|\text{softmax}\left(\frac{z_s}{T}\right)\right) \tag{2}$$

Unlike pure cross-entropy approaches that prioritize answer correctness or pure KL methods that risk overfitting to teacher idiosyncrasies, our hybrid approach achieves superior reasoning transfer with manageable computational requirements.

Two key innovations enhance our framework's performance on mathematical tasks:

- **Adaptive temperature scheduling**: Beginning with higher values to learn broad reasoning strategies before transitioning to lower temperatures that enhance precision.

- **Reasoning-focused weighted masking**: Placing greater emphasis on reasoning steps rather than problem statements to prioritize mathematical deduction processes.

The distilled model achieved 72.1% accuracy on AIME'24—a substantial improvement over the base model's 50.0%, though token usage remained high at approximately 31,764 tokens per problem. This observation motivated our subsequent GRPO implementation to improve efficiency while preserving reasoning capabilities.

5 Group Relative Policy Optimization

Standard reinforcement learning techniques face prohibitive memory constraints when applied to large language models for mathematical reasoning. We address this challenge through a memory-optimized implementation of Group Relative Policy Optimization (GRPO) with a novel zero KL penalty formulation.

5.1 Policy Optimization Challenges

Proximal Policy Optimization (PPO) [Schulman et al., 2017] has become the standard method for aligning language models through reinforcement learning. However, its actor-critic architecture requires maintaining a separate value network alongside the policy model, creating significant memory overhead that becomes prohibitive for models with tens of billions of parameters—a critical limitation for mathematical reasoning where model size strongly correlates with reasoning ability.

5.2 GRPO and Zero KL Penalty Innovation

Group Relative Policy Optimization [Shao et al., 2024] addresses these limitations by eliminating the separate critic network, instead deriving advantage estimates from grouped sample returns. Our implementation takes a single gradient step per batch of trajectories with a simplified loss function.

The key innovation in our approach is setting the KL divergence penalty coefficient β to zero. While conventional wisdom in RLHF advocates for a positive KL penalty to prevent policy divergence, our analysis revealed that for mathematical reasoning tasks, allowing greater distributional shift from the initial model produced superior outcomes. With advantage normalization and online RL, the loss ultimately simplifies to $J(\theta) = -\beta D_{KL}(\pi_\theta || \pi_{ref})$, which vanishes when $\beta = 0$.

This enables us to completely eliminate the reference model from memory while counter-intuitively improving mathematical reasoning performance. The advantage estimation uses sample group normalization where $\hat{A}_{i,t} = \frac{r_i - \text{mean}(r)}{\text{std}(r)}$, preserving the stability benefits of PPO's approach.

5.3 Systems-Level Implementation

Our memory-optimized implementation enables efficient training of 32B parameter models on just 8 GPUs through several technical innovations:

- **Distributed inference**: Parallel execution across 8 vLLM engines processing unique prompt batches
- **CPU-GPU memory orchestration**: Strategic offloading of vLLM engines during policy updates
- **Parameter-efficient sharding**: DeepSpeed ZeRO-3 with CPU offloading of optimizer states
- **Reference model elimination**: Zero KL penalty approach removes the need for a reference model

These optimizations create an efficient training cycle that enables fine-tuning of models that would otherwise require substantially more computational resources. The resulting models demonstrate more concise reasoning patterns, using 6.3× fewer tokens while maintaining or improving reasoning accuracy.

6 Multi-agent PRM Reranking

While our GRPO approach optimizes model parameters for mathematical reasoning, individual solution attempts remain susceptible to errors. We developed a multi-agent framework with preference-based reranking to enhance solution quality through strategic diversity.

Our framework coordinates multiple solution-generating agents with varied sampling parameters (temperature, top-p) to create solution diversity, while a Preference Reward Model (PRM) evaluates and ranks these solutions based on quality criteria:

- **Solution generation and verification**: Multiple model instances generate diverse solutions that capture different mathematical approaches, with both self-reflection and cross-verification to identify potential errors.
- **Quality-based ranking**: The PRM evaluates solutions across key dimensions (answer correctness, reasoning quality, technique appropriateness, and clarity), scoring each solution to enable optimal selection without requiring explicit mathematical rules.

This approach is particularly valuable for competition-level mathematics problems that admit multiple valid solution strategies. By generating and evaluating diverse approaches, our system selects the most appropriate technique for each specific problem.

To address occasional instabilities in models after GRPO training, we implemented a targeted post-GRPO stabilization phase focusing on solution consistency, advanced technique application, and verification integration. This counterbalances the exploration encouraged during reinforcement learning.

The multi-agent framework significantly improved performance, achieving 79.9% accuracy on AIME'24 compared to 76.7% for single-inference approaches—a 3.2 percentage point improvement. These gains were particularly pronounced for problems requiring complex multi-step reasoning, confirming that the multi-agent approach effectively addresses single-solution limitations while maintaining token efficiency.

7 Experiments

We conducted comprehensive experiments to evaluate our three-stage approach across different model scales and to isolate the contribution of each pipeline component.

7.1 Experimental Setup

7.1.1 Models and Baselines

We experimented with three model scales and compared against five competitive baselines.

Model	Size	Description
Our Models		
Qwen2.5-1.5B (Ours)	1.5B	Resource-efficient variant
Qwen2.5-7B (Ours)	7B	Medium-scale variant
Qwen2.5-32B (Ours)	32B	Primary model
Baselines		
Qwen 32B Base	32B	Foundation model
Qwen 32B Instruct	32B	Instruction-tuned variant
Qwen2.5-Math-72B	72B	Mathematical reasoning model
Qwen2.5-Math-72B w/ TIR	72B	Tool-Integrated Reasoning
Qwen-R1-Distilled-32B	32B	Distilled from DeepSeek-R1

Table 1: Models used in our experiments

7.1.2 Training Configuration

All experiments were conducted on 8 NVIDIA A100 80GB GPUs.

Configuration	1.5B	7B	32B
Learning rate	2e-6	2e-6	2e-6
Batch size	64	256	384
KL penalty (β)	0.001	0	0
Max gradient norm	1.0	0.5	0.5
Context length	8192	8192	16384
Temperature	0.7	0.7	0.7
Top-p	0.95	0.95	0.95

Table 2: Training hyperparameters across model scales

7.2 Progressive Experimental Validation

7.2.1 Initial GRPO Validation (1.5B Scale)

Our preliminary experiment used Qwen-1.5B-Instruct on the GSM8K benchmark with a two-component reward function:

$$R = \begin{cases} 0.2, & \text{if the output uses correct format} \\ 1.0, & \text{if the output is correct} \end{cases} \tag{3}$$

This initial validation confirmed the efficacy of our zero KL penalty approach while establishing baseline improvements at smaller model scales.

7.2.2 Mid-scale Implementation (7B Parameters)

Building on our initial findings, we implemented our full pipeline with DeepSeek-R1-Distill-Qwen-7B using an expanded batch size of 256 completions per update. This experiment verified the scalability of our memory-optimized GRPO implementation and demonstrated the importance of larger batch sizes for mathematical reasoning tasks.

7.2.3 Primary Model Optimization (32B Scale)

Our primary model combined lessons from previous experiments with several enhancements:

- **Expanded exploration space**: Increased rollout_batch_size to 48
- **Enhanced reward formulation**: Added mathematical validity checks
- **Curriculum optimization**: Progressive difficulty increases
- **Extended context window**: Utilized 16,384 token context

These refinements enabled more effective learning while maintaining computational efficiency.

Evaluation Protocol We evaluated our models on benchmarks designed to assess different aspects of mathematical reasoning. Performance was measured using exact match accuracy for final answers, with partial credit assigned for multi-part problems.

Benchmark	Size	Description
AIME'24	15 problems	Competition-level problems from AIME
OpenR1-Math	1,000 problems	Holdout from OpenR1-Math-220K

Table 3: Evaluation benchmarks

7.2.4 Multi-agent Configuration

For the multi-agent framework evaluation, we used 4 parallel inference passes with strategically varied sampling parameters.

Parameter	Agent 1	Agent 2	Agent 3	Agent 4
Temperature	0.2	0.4	0.6	0.8
Top-p	0.85	0.90	0.92	0.95
Max tokens	4096	4096	4096	4096

Table 4: Multi-agent inference configuration

The PRM model was fine-tuned on paired solutions with preference annotations, using a base Qwen2.5-7B model with 1000 training steps.

8 Results

We evaluated our three-stage mathematical reasoning framework across multiple benchmarks, focusing on both performance accuracy and computational efficiency.

8.1 AIME'24 Benchmark Performance

On the challenging AIME'24 competition-level mathematics benchmark, our approach achieved state-of-the-art results while dramatically reducing computational requirements.

Model	AIME'24 Accuracy	Tokens Used
Qwen 32B Base	50.0%	32000
Qwen 32B Instruct	26.7%	32000
Qwen2.5-Math-72B	30.0%	32000
Qwen2.5-Math-72B (TIR)	40.0%	32000
Qwen-R1-Distilled	73.3%	32000
Our Model (Single)	76.7%	5048
Our Model (Multi-agent)	**79.9%**	5048 × 4

Table 5: Performance comparison showing superior accuracy and reduced token usage.

Our single-inference model achieves 76.7% accuracy using only 5048 tokens per problem—a 6.3× reduction in computational requirements compared to baselines. The multi-agent approach further improves accuracy to 79.9%, outperforming even the strongest baseline (Qwen-R1-Distilled at 73.3%) while maintaining the same token efficiency per agent.

8.2 Component Contribution Analysis

To quantify the impact of each pipeline component, we performed systematic ablation studies:

Model Configuration	AIME'24 Accuracy	Avg. Tokens
Full Pipeline (multi-agent)	79.9%	5048 × 4
Without Multi-agent PRM	76.7%	5048
Without GRPO (Distill only)	72.1%	31764
Without Distillation (Base)	50.0%	32000

Table 6: Ablation results demonstrating incremental contribution of each component.

Each stage of our pipeline contributes meaningfully to the final performance: distillation establishes mathematical reasoning foundations (+22.1 percentage points over the base model); GRPO significantly improves token efficiency while further enhancing accuracy (+4.6 points with 84% token reduction); and multi-agent PRM provides the final performance boost (+3.2 points) through verification and reranking.

8.3 Scaling and Generalization Analysis

Our approach demonstrates consistent performance improvements across model scales, with larger models benefiting more substantially from GRPO training:

Model	AIME'24	Holdout
Qwen2.5-1.5B GRPO	10.0%	—
Qwen2.5-7B GRPO	16.7%	—
Qwen2.5-32B Base	50.0%	54.6%
Qwen2.5-32B GRPO (checkpoint)	—	58.5%
Qwen2.5-32B GRPO (final)	**76.7%**	**63.1%**

Table 7: Performance across model scales and on holdout evaluation.

Evaluation on a 1,000-problem holdout set from OpenR1-Math-220K demonstrates that our improvements generalize beyond the competition-specific AIME benchmark. We observed progressive accuracy gains throughout training, with a notable improvement after extending the context window from 8K to 16K tokens at checkpoint-350, highlighting the importance of sufficient context for complex mathematical reasoning.

8.4 Token Efficiency Mechanisms

Our models achieve remarkable token efficiency through several learned behaviors:

- **Adaptive allocation**: Using fewer tokens for simpler problems while allocating more to complex ones requiring detailed reasoning
- **Redundancy elimination**: Removing unnecessary explanation steps without sacrificing solution validity
- **Technique optimization**: Focusing on the most relevant mathematical approaches for each problem type

This adaptive token usage resulted in 2.1× faster inference compared to baseline models—a critical advantage for practical applications where inference cost and latency matter. The efficiency gains are particularly noteworthy considering they occur without compromising—and in fact improving—mathematical reasoning accuracy.

9 Ablation Studies and Analysis

We conducted comprehensive analyses to understand component contributions, error patterns, and efficiency-accuracy trade-offs in our mathematical reasoning framework.

9.1 Component Contribution Analysis

Beyond basic ablation tests, we examined how each pipeline component influences overall performance through targeted parameter variations:

- **Distillation objective balance**: Varying the KL+CE loss weighting parameter α revealed an optimal mid-range value ($\alpha \approx 0.5$), balancing reasoning structure transfer and answer accuracy. Low α values preserved solutions but compromised reasoning coherence, while high values maintained reasoning structure but reduced answer precision.

- **GRPO KL penalty impact**: Our zero KL penalty approach not only reduced memory requirements by 38% but also accelerated convergence by 22% compared to traditional PPO implementations with standard KL penalties. This optimization proved particularly beneficial for token efficiency improvements.

- **Multi-agent scaling**: Performance increased logarithmically with agent count, showing substantial gains from 1 to 4 agents (+3.2 percentage points) but diminishing returns beyond that point (+0.7 points from 4 to 8 agents), confirming our 4-agent configuration as near-optimal for the performance-compute trade-off.

9.2 Error Analysis

Detailed examination of model errors on the AIME'24 benchmark revealed structured patterns:

The error distribution demonstrates domain-specific challenges, with geometry problems showing the highest error rate. Across all domains, conceptual misunderstandings (43%) outweighed computational errors (27%) and incomplete reasoning (30%), suggesting that future improvements should prioritize enhancing conceptual representation rather than calculation capabilities.

9.3 Tool Integration and Cross-Domain Analysis

We evaluated both Tool-Integrated Reasoning (TIR) and cross-domain generalization capabilities:

Model	AIME'24	Cross-Domain
Qwen2.5-1.5B	10.0%	18.1%
Qwen2.5-32B (Ours)	**76.7%**	**60.2%**

Table 8: Performance on AIME'24 and cross-domain tasks (averaged across calculus and abstract algebra).

Tool-Integrated Reasoning shows the most significant relative improvement on smaller models, suggesting external tools can partially compensate for limited model capacity. Meanwhile, our 32B model demonstrates substantial cross-domain generalization capabilities, though with expected performance degradation on domains not specifically targeted during training.

9.4 Efficiency-Accuracy Trade-off Analysis

We systematically explored the relationship between token limit and reasoning accuracy by varying maximum allowed tokens during inference:

- **Performance inflection points**: Accuracy increases steeply up to 4000 tokens (72.1%), moderately to 6000 tokens (77.3%), and plateaus beyond 8000 tokens (77.9%).

- **Optimal operating range**: The 4000-6000 token range represents the sweet spot for balancing accuracy and efficiency, with our chosen 5048 token limit capturing 98.4% of maximum performance at less than 20% of the computational cost of standard 32K approaches.

- **Problem-adaptive allocation**: Token usage analysis reveals that our model adaptively allocates tokens based on problem complexity (r=0.72 correlation between problem difficulty and token usage), demonstrating efficient resource utilization.

These analyses confirm that our approach effectively resolves the fundamental efficiency-accuracy trade-off in mathematical reasoning, with particular benefits for deployment scenarios where computational resources are constrained but reasoning quality cannot be compromised.

10 Conclusion

Our work establishes a novel framework for mathematical reasoning in large language models that successfully resolves the longstanding tension between reasoning quality and computational efficiency. By synergistically combining reasoning capability distillation, memory-optimized GRPO, and multi-agent PRM reranking, we achieved state-of-the-art performance on competition-level mathematics while dramatically reducing computational requirements.

Our models demonstrate unprecedented token efficiency, achieving a 6.3× reduction in token usage while improving accuracy on the challenging AIME'24 benchmark to 76.7

10.1 Impact and Future Work

The advancements presented in this work have significant implications beyond benchmarks and open several promising research directions:

- **Democratizing advanced reasoning**: Our approach makes sophisticated mathematical reasoning more accessible across diverse deployment environments, particularly benefiting educational applications where response time significantly impacts student engagement and learning outcomes. The substantial reduction in token usage also translates to meaningful energy savings at scale.

- **Extending to other domains**: This framework can be adapted to other complex reasoning tasks such as scientific problem-solving and algorithmic reasoning. Future work could explore integration with iterative refinement mechanisms like Chain-of-Draft [Xu et al., 2025] and specialized tool integration for precise computation tasks.

- **Scaling and generalization**: Further research should investigate how these techniques generalize across model scales from sub-billion to hundred-billion parameter ranges and across different mathematical domains.

In summary, our three-stage approach establishes a new paradigm for mathematical reasoning in LLMs that effectively balances performance and efficiency, making advanced reasoning capabilities more practical for real-world applications.

Limitations

Despite the significant improvements demonstrated in our work, several limitations remain:

Our current model shows varying performance across mathematical subfields, with relative strengths in algebra and number theory but weaker performance in geometry and probability. This suggests a need for more balanced training data across mathematical domains.

While multiagent approaches improve solution quality, they sometimes converge to similar reasoning paths, which limits the diversity of solution approaches. Developing techniques to encourage more diverse reasoning styles could improve robustness.

More research is needed to understand how our approach scales to even larger models (100B+ parameters) and whether the token efficiency gains continue to increase with scale.

Our evaluation focused primarily on competition-level mathematics problems. Real-world mathematical applications may present different challenges that require customized training or fine-tuning approaches.

Ethics Statement

Our work aims to advance mathematical reasoning capabilities in AI systems, which has broad applications in the fields of education, scientific research, and engineering. By improving both accuracy and computational efficiency, we make advanced mathematical reasoning more accessible.

The models developed in this work do not present significant ethical concerns beyond those common to all large-language models. We have carefully curated training data to focus on well-established mathematical content and problems. The models are used to solve mathematical problems rather than to generate text that might contain harmful biases or misinformation.

The reduced token usage in our models has positive environmental implications by decreasing the computational resources required for inference, potentially leading to lower energy consumption when deployed at scale.

References

DeepSeek AI. Deepseek-r1: Human-level reasoning using large language models with tools. *arXiv preprint*, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213, 2022.

Shaoguang Li, Zheng Hu, Yingxue Wang, Yang Gao, Xiaoyu Chen, Wen Bian, Yujie Wang, Lei Wang, Jun Liu, Zhiheng Jin, et al. Numinamath 1.5: Scaling data and performance for mathematical reasoning. *arXiv preprint arXiv:2405.18415*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Xiang Luo, Jiangjie Chen, Cheng Wang, Yinya Wei, and Zhenwen Zhang. The agentica project: Scaling reinforcement learning for mathematical reasoning. *arXiv preprint*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Junlong Shao, Shenyu Zheng, Peter Jin, Zhengyu Xu, Yuhui Wang, Yang Bei, Ji-Rong Wen, and Ruoxi Jia. Group relative policy optimization for efficient learning in human feedback alignment. *arXiv preprint arXiv:2402.10632*, 2024.

Jiacheng Wang, Rui Han, Shisheng Liu, Linjun Liao, Ke Lin, Wenxuan Wang, Zhiyuan Deng, Xiyao Li, Yinliang Wu, Hongxia Gao, Weiran Ma, et al. Math-shepherd: A language model for mathematical reasoning with function calling. *arXiv preprint arXiv:2312.08935*, 2023a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations*, 2023b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.

Wayne Wu, Denny Zhou, and Susan Zhang. Collaborative mathematics: An investigation of collective intelligence in mathematics problem solving. *arXiv preprint arXiv:2312.17374*, 2023.

Alex Xu, Haotian Ding, Shizhan Cheng, Qing Gao, Liam Young, Anca Dragan, and Percy Liang. Chain-of-draft: Solving complex problems with iterative refinement. *arXiv preprint*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Zhenwen Zhang, Yunhao Zhao, Yongqiang Wu, Zhehao He, Xiang Luo, Gengchen Wang, Haolan Xu, Binqiang Jiao, Jidong Fu, Yanyan Lan, et al. Equation-of-thought prompting: Empowering mathematical reasoning in large language models. *arXiv preprint arXiv:2309.13177*, 2023.