# Enhancing Mathematical Reasoning in Large Language Models through Reasoning Distillation, GRPO, and Multi-agent PRM Reranking

**Abhinav Agarwal**
Department of Computer Science
Stanford University
abhinav4@stanford.edu

**Carlo Baronio**
Department of Mathematics
Stanford University
cbaronio@stanford.edu

**Shree Reddy**
Department of Computer Science
Stanford University
shreered@stanford.edu

**Shubhra Mishra**
Department of Computer Science
Stanford University
shubhra@stanford.edu

## Abstract

We present a comprehensive approach to enhancing mathematical reasoning in large language models through the combination of distillation, Group Relative Policy Optimization (GRPO), and multi-agent verification. Using a curated dataset of 387K mathematical problems, we first transfer reasoning capabilities from teacher to student models, then apply memory-optimized GRPO with zero KL penalty for efficient fine-tuning. Our approach achieves 76.7% accuracy on the challenging AIME'24 benchmark using just 5048 tokens per problem—a 6.3× reduction compared to baseline models requiring 32K tokens. The multi-agent framework further improves performance to 79.9%. By systematically addressing both accuracy and efficiency constraints, our method makes advanced mathematical reasoning more accessible for real-world applications. Ablation studies demonstrate that each component contributes meaningfully to the final results, with consistent improvements across model scales from 1.5B to 32B parameters.

## 1 Introduction

The AIMO Progress Prize challenge represents a significant milestone in advancing AI capabilities for mathematical reasoning. Success in this competition requires solving complex mathematical problems at the national olympiad level, demanding sophisticated reasoning, creativity, and precise computation. Existing LLMs struggle with consistency and accuracy when solving advanced mathematical problems, while also using excessive computational resources.

Mathematical reasoning presents unique challenges for LLMs compared to general applications. While language understanding has advanced significantly, mathematical problem-solving requires domain knowledge, multi-step logical reasoning, and symbolic consistency throughout long solution processes. Key challenges include (1) maintaining coherent reasoning across complex multi-step problems, (2) balancing computational efficiency with solution quality, and (3) achieving consistent performance across diverse problem types.

In this work, we propose a comprehensive approach combining distillation, reinforcement learning, and multi-agent verification to create models that reason more effectively while using computational resources more efficiently. Our method achieves 76.7% accuracy on the challenging AIME'24

benchmark with just 5048 tokens per problem—a 6.3x reduction in token usage compared to baseline models. When applied with our multi-agent framework, accuracy increases to 79.9%.

Our key contributions include:

- A curated dataset of 387K mathematical problems focused on olympiad-level content
- An optimized knowledge distillation technique transferring reasoning abilities from larger to smaller models
- A memory-efficient Group Relative Policy Optimization (GRPO) implementation enabling scaling to 32B parameter models
- A multi-agent PRM reranking framework for generating and selecting high-quality solutions
- State-of-the-art performance on AIME'24 benchmarks with significantly reduced computational requirements

## 2 Related Work

### 2.1 Mathematical Reasoning in LLMs

Recent advances in mathematical reasoning capabilities of large language models have shown promising results but still face significant challenges. Models like DeepSeek-Math [Shao et al., 2024] and MAmmoTH [Zhang et al., 2023] use specialized pretraining and fine-tuning to improve mathematical capabilities, while approaches like tool-augmented reasoning [Jiang et al., 2023] provide external computational aids. Traditional approaches using supervised finetuning often capture surface patterns of solutions without developing true reasoning abilities. Our work builds on these foundations while addressing key limitations in token efficiency and reasoning consistency.

### 2.2 Reinforcement Learning for Reasoning Tasks

Prior work has explored reinforcement learning approaches like Proximal Policy Optimization (PPO) [Schulman et al., 2017] to improve mathematical reasoning. However, these approaches often require extensive computational resources and struggle with the sparse reward landscape of mathematical problem-solving. Alternative approaches include Direct Preference Optimization [Rafailov et al., 2023] and Constitutional Reinforcement Learning [Yang et al., 2023], which offer different paradigms focused on policy improvement through preference modeling. Our GRPO implementation builds on the approach introduced by Shao et al. [2024] but with novel optimizations for memory efficiency and scalability to larger models.

### 2.3 Multi-agent and Verification Approaches

Self-consistency [Wang et al., 2023b] improves reasoning by generating multiple solutions and selecting the most common answer, while verification-based approaches [Lightman et al., 2023] explicitly evaluate solution correctness. However, these approaches typically rely on ensemble methods that increase inference costs substantially. The multi-agent framework in Math-Shepherd [Wang et al., 2023a] demonstrated the benefits of specialized agent roles. Our approach combines these ideas with a PRM reranking mechanism, enabling more robust solution selection without relying solely on majority voting, while maintaining computational efficiency.

### 2.4 Knowledge Distillation for Specialized Domains

Knowledge distillation has emerged as an effective technique for transferring capabilities from larger to smaller models, with approaches like sequence-level knowledge distillation [Kim and Rush, 2016] and response-based distillation [Hinton et al., 2015]. Chain-of-Thought distillation [Li et al., 2023] specifically targets reasoning capabilities, which we adapt for mathematical domains with our hybrid KL+CE approach. Distillation provides an efficient way to transfer reasoning abilities while controlling computational costs during both training and inference.

## 3 Dataset Creation and Curation

To train our models effectively, we created a comprehensive dataset combining high-quality existing resources with newly generated problems, focusing on competition-level mathematics.

### 3.1 Dataset Composition and Processing

Our final dataset comprised 387K problems from three primary sources:

- **OpenR1-Math-220K** (55%): Contains 2-4 reasoning traces per problem, providing diverse solution approaches
- **Pipeline-generated content** (37.5%): 150K problems with optimized solution generation
- **Bespoke-Stratos-17K** (7.5%): High-quality problems with detailed step-by-step solutions

We developed a customized processing pipeline that:

- Removed overlapping problems to prevent data contamination
- Filtered out K-12 level problems to focus on challenging content
- Selected problems based on validity tags and mathematical concepts
- Ensured balanced representation across difficulty levels

### 3.2 Solution Generation and Quality Assessment

For approximately 150K problems, we generated new solutions using an optimized pipeline that reduced processing time from 10 min/batch to 3 min/batch while maintaining solution quality. Our multi-stage verification system included:

- Model-based verification using OpenAI-4o-mini as a judge to evaluate mathematical validity
- Automatic filtering of problems with invalid tags
- Quality checks for solution correctness, reasoning validity, and format consistency

This process rejected approximately 15% of initially generated solutions, ensuring high quality throughout the dataset. The resulting collection provides both breadth and depth in mathematical concepts, with particular emphasis on olympiad-level content for alignment with competition standards.

## 4 Reasoning Capability Distillation

We implemented a specialized knowledge distillation framework to transfer mathematical reasoning capabilities from larger, more capable models to our target models. This approach enables the student model to learn not just the outputs but the entire reasoning process of the teacher model.

### 4.1 Knowledge Distillation Framework

Our distillation process uses DeepSeek R1 (teacher) and Qwen2.5 models (student) with Chain-of-Thought (CoT) distillation. The objective function combines two components:

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

where $\mathcal{L}_{\text{KL}}$ is the KL divergence loss between the softened logits of the teacher and student models, and $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss between the student's predictions and the ground truth.

Our KL+CE hybrid approach balances accuracy with reasoning fidelity, providing distinct advantages over alternative methods:

| Method | Reasoning Transfer | Efficiency | Complexity |
|---|---|---|---|
| Pure CE | Low | High | Low |
| Pure KL | Medium | Medium | Medium |
| KL+CE (Ours) | High | Medium | Medium |

### 4.2 Key Optimizations

Our implementation includes several optimizations that significantly improve the quality and efficiency of the distillation process:

- **Adaptive temperature scheduling**: Starting with higher temperatures for general pattern learning before transitioning to lower temperatures for precision
- **Weighted masking**: Placing greater emphasis on reasoning steps rather than problem statements
- **Training efficiency**: Automatic mixed precision (AMP) with gradient accumulation

The adaptive temperature approach provides significant benefits: during early training (high $T$), the student learns broad reasoning patterns from softened teacher distributions, while in late training (low $T$), focus shifts to precise token selection and accurate final answers.

This distillation process provides a strong foundation for the subsequent reinforcement learning stage, effectively transferring the reasoning capabilities while maintaining computational efficiency.

## 5 Memory-Optimized GRPO Training

GRPO (Group Relative Policy Optimization) offers a more efficient alternative to traditional PPO by eliminating the need for a separate critic network, instead estimating baseline rewards from grouped sample returns. This significantly reduces memory requirements while maintaining stable policy optimization.

### 5.1 Efficient GRPO Implementation

We implemented GRPO with several optimizations for large language model training. The core objective function is:

$$J(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{[\pi_\theta(o_{i,t}|q,o_{i,<t})]_\nabla}\hat{A}_{i,t} - \beta D_{KL}(\pi_\theta\|\pi_{ref})\right] \tag{2}$$

where $\hat{A}_{i,t}$ represents normalized advantages and $[]_\nabla$ denotes stop gradient.

Our key innovation was setting $\beta = 0$ to eliminate the KL divergence penalty term. This approach:

- Allows faster learning, especially with format-based rewards
- Removes the need to store the reference model in memory
- Reduces computational requirements by eliminating reference model forward passes

### 5.2 Systems Optimizations

We implemented several system-level optimizations to enable efficient training of large models:

- Distributed inference across 8 vLLM engines, each with unique prompt sets
- Memory management via CPU offloading of vLLM engines during policy updates
- DeepSpeed ZeRO-3 for parameter, gradient, and optimizer state partitioning
- Zero KL penalty ($\beta = 0$) to reduce memory requirements further

This optimized implementation allowed us to efficiently train models up to 32B parameters. Our approach creates a streamlined RL training pipeline that maintains high-quality optimization while significantly reducing computational overhead compared to standard PPO implementations.

# 6 Multi-agent PRM Reranking

To enhance solution quality and diversity, we implemented a multi-agent framework that generates multiple solution attempts, evaluates them, and selects the most promising ones. This approach allowed us to increase AIME'24 accuracy from 76.7% (single inference) to 79.9% (multi-agent framework).

## 6.1 Framework Components

Our multi-agent system consists of four integrated components:

- **Diverse Solution Generation**: Multiple agents with varied parameters produce different solution approaches
- **Self-reflection and Verification**: Each solution undergoes validation for mathematical correctness
- **Preference Reward Model (PRM)**: Ranks solutions based on quality metrics including correctness, clarity, and technique appropriateness
- **Solution Selection**: Identifies the highest-quality solution through reward-based scoring

## 6.2 Implementation Advantages

This approach offers several benefits over single-inference methods:

- **Increased robustness**: Mitigates individual solution failures through ensemble approaches
- **Method diversity**: Captures multiple valid solution strategies for complex problems
- **Consistency verification**: Cross-validates solutions against each other
- **Quality assurance**: PRM ensures selection of the highest-quality reasoning path

Our implementation efficiently manages the generation-evaluation-selection process through parallel inference and a specialized post-GRPO fine-tuning phase. This fine-tuning further stabilizes the model's reasoning capabilities after exploration during RL training, helping it adapt to competition-style problems.

# 7 Experiments and Results

We conducted extensive experiments across different model scales to evaluate our approach and identify key factors in training effective mathematical reasoning models.

## 7.1 Key Training Observations

Our experiments revealed several important insights about GRPO training for mathematical reasoning:

- **Context window impact**: After approximately 320 training steps with 7B models, we observed performance degradation due to truncated completions on more difficult problems. Expanding the context window from 8192 to 16384 tokens restored and improved performance.
- **Token efficiency learning**: Models initially learned to use tokens more efficiently during training, with completion lengths decreasing, before selectively increasing token usage for harder problems to improve accuracy.

- **Reward structure**: A combined reward approach using both format (0.2) and correctness (1.0) components provided effective training signals. The format reward encouraged adherence to structured reasoning while the correctness reward focused on accurate final answers.

### 7.2 Model Scaling and Performance

We extended our GRPO implementation across multiple model scales, observing consistent improvements with increasing model size:

| Model | AIME'24 Accuracy |
|---|---|
| Qwen2.5-1.5B GRPO | 10.0% |
| Qwen2.5-7B GRPO | 16.7% |
| Qwen2.5-32B Base | 50.0% |
| Qwen2.5-32B Instruct | 26.7% |
| Qwen2.5-32B GRPO | 76.7% |

The Qwen2.5-32B GRPO model showed a remarkable 26.7% accuracy gain over the base model

- Larger batch sizes (rollout_batch_size = 48) for better exploration
- Curriculum learning to gradually increase problem difficulty
- Extended context window (16384 tokens) throughout training
- Refined reward function with weighted combinations

The final 32B model achieved 76.7% accuracy on AIME'24 with single-pass inference, increasing to 79.9% with our multi-agent approach using 4 inference passes.

## 8 Results

### 8.1 Benchmark Performance and Ablation Studies

Our final 32B model achieved 76.7% accuracy on the challenging AIME'24 benchmark with just 5048 tokens per problem, while our multi-agent approach reached 79.9% using 4 inference passes. This significantly outperforms strong baselines including Qwen 32B Base (50.0%), Qwen2.5-Math-72B (30.0%), and Qwen-R1-Distilled-32B (73.3%)—all of which require 32K tokens per problem.

| Model Configuration | AIME'24 Accuracy | Avg. Tokens Used |
|---|---|---|
| Full Pipeline (with multi-agent) | 79.9% | $5048 \times 4$ |
| Without Multi-agent PRM | 76.7% | 5048 |
| Without GRPO (Distillation only) | 72.1% | 31764 |
| Without Distillation (Base model) | 50.0% | 32000 |

Our ablation studies confirm that each component contributes meaningfully to the final performance:

- Distillation provides the foundation of mathematical reasoning capabilities (+22.1%)
- GRPO significantly improves token efficiency while enhancing accuracy (+4.6%)
- Multi-agent PRM adds the final performance boost (+3.2%)

We observed similar relative improvements with 7B models, though with lower absolute performance (65.2% accuracy with 5137 tokens versus 58.7% for the baseline using 32K tokens).

### 8.2 Token Efficiency Analysis

A key finding from our experiments is the model's ability to adaptively optimize token usage:

- Using fewer tokens for simpler problems

- Allocating more tokens to complex problems requiring detailed reasoning
- Removing redundant explanation steps
- Focusing on the most relevant mathematical techniques for each problem

This adaptive token allocation resulted in a 6.3× reduction in overall token usage compared to baseline models while improving accuracy. The trained models also demonstrated 2.1× faster inference on our test dataset by learning to spend Chain-of-Thought tokens only where they provide maximum benefit.

# 9 Performance Comparison

We conducted a comprehensive evaluation of our model against state-of-the-art baselines on the AIME'24 benchmark, measuring both accuracy and token efficiency.

## 9.1 Comparative Results

Our approach achieved superior results compared to larger and more specialized models:

| 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% | $5048 \times 4$ |

This comparison demonstrates that our approach not only achieves higher accuracy than specialized models like Qwen2.5-Math-72B with Tool-Integrated Reasoning (40.0%), but does so with significantly fewer tokens per inference.

## 9.2 Efficiency-Performance Trade-offs

Our work highlights important trade-offs between model size, computational efficiency, and reasoning capability:

- Our single-inference model (76.7% accuracy) provides the best balance of performance and efficiency
- The multi-agent approach (79.9% accuracy) achieves peak performance at the cost of additional computation
- Even our smaller models (7B parameters) outperform larger baselines when trained with our pipeline

These results demonstrate that thoughtful training approaches can be more effective than simply scaling model size, especially when computational efficiency is important.

# 10 Tool-Integrated Reasoning Analysis

## 10.1 TIR Performance Across Model Sizes

Tool-Integrated Reasoning (TIR) represents an important advancement in mathematical problem-solving capabilities for language models. By leveraging external computational tools, models can overcome their inherent limitations in performing precise calculations or applying specialized algorithms.

| Model | Standard | With TIR | Improvement |
|---|---|---|---|
| Qwen2.5-Math-1.5B | 10.0% (3/30) | 23.3% (7/30) | +13.3% |
| | | 30.0% $(9/30)_{\text{maj@64}}$ | +20.0% |
| | | 60.0% $(18/30)_{\text{rm@64}}$ | +50.0% |
| | | 30.0% $(9/30)_{\text{maj@256}}$ | +20.0% |
| | | 63.3% $(19/30)_{\text{rm@256}}$ | +53.3% |
| Qwen2.5-Math-7B | 16.7% (5/30) | 20.0% (6/30) | +3.3% |
| | | 43.3% $(13/30)_{\text{maj@64}}$ | +26.6% |
| | | 70.0% $(21/30)_{\text{rm@64}}$ | +53.3% |
| | | 46.7% $(14/30)_{\text{maj@256}}$ | +30.0% |
| | | 70.0% $(21/30)_{\text{rm@256}}$ | +53.3% |
| Qwen2.5-Math-72B | 30.0% (9/30) | 40.0% (12/30) | +10.0% |
| | | 46.7% $(14/30)_{\text{maj@64}}$ | +16.7% |
| | | 60.0% $(18/30)_{\text{rm@64}}$ | +30.0% |
| | | 53.3% $(16/30)_{\text{maj@256}}$ | +23.3% |
| | | 63.3% $(19/30)_{\text{rm@256}}$ | +33.3% |

Figure 1: Tool-Integrated Reasoning performance across model sizes. Subscripts indicate the evaluation method: maj = majority voting, rm = reward model scoring, and the number indicates inference budget.

### 10.2 Key Findings

Our analysis of TIR performance across model scales reveals several important patterns:

1. **Universal Improvement**: TIR consistently improves performance across all model sizes, with even the smallest 1.5B model showing substantial gains.
2. **Smaller Models Benefit More**: The relative improvement from TIR is more pronounced for smaller models (percentage-wise), suggesting TIR can partially compensate for limited parameter count.
3. **Scaling with Inference Budget**: Increasing the inference budget from single inference to majority voting (maj@64/256) and reward model scoring (rm@64/256) consistently improves performance.
4. **Reward Model Superiority**: Across all model sizes, reward model scoring significantly outperforms majority voting as a selection mechanism, highlighting the importance of solution quality assessment over simple consensus.

### 10.3 Comparative Analysis with Our Approach

While TIR provides significant improvements, particularly for smaller models, our GRPO-trained 32B model achieves 76.7% accuracy without requiring specialized external tools, and 79.9% with our multi-agent approach. This suggests that:

- The reasoning capabilities learned through our pipeline generalize better than tool reliance
- For smaller models (1.5B, 7B), TIR remains an effective strategy to boost performance
- For larger models where computational efficiency is critical, our token-efficient GRPO approach provides superior results
- The optimal approach may involve combining both strategies: GRPO-trained models that can selectively leverage external tools when necessary

These findings suggest different deployment strategies depending on the available computational resources and accuracy requirements.

## 11 Conclusions

Our work demonstrates that combining distillation, GRPO, and multi-agent verification creates a powerful framework for enhancing mathematical reasoning in large language models. We achieved 79.9% accuracy on AIME'24 benchmark with significantly reduced computational requirements compared to baseline models.

Key findings from our research include:

- Reasoning capabilities can be effectively transferred from larger to smaller models through our specialized distillation techniques
- Memory-optimized GRPO offers an efficient approach to reinforcement learning for mathematical reasoning, eliminating the need for a separate critic
- Multi-agent frameworks provide significant accuracy improvements by leveraging solution diversity and verification
- Token efficiency can be improved 6.3× without sacrificing accuracy through our integrated approach

Our approach demonstrates that thoughtful training methodologies can be more effective than simply scaling model size. By optimizing for both accuracy and computational efficiency, we make advanced mathematical reasoning more accessible for practical applications.

Future work could explore extending our approach to other reasoning domains beyond mathematics, developing hybrid approaches combining our token-efficient models with specialized tools, and investigating the emergent properties of multi-agent reasoning systems with more specialized roles.

Our code and models are available at `https://github.com/abhinav30219/math_reasoner` and `https://huggingface.co/abhinav302019/Math-Reasoning-Model-32B`.

## References


Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Albert Q. Jiang, Noah Weber, Aniruddh Srivastava, Danny Driess, Po-Sen Huang, Joao Sedoc, Christopher D. Manning, Gregory Valiant, Chelsea Finn, Dylan Hadfield-Menell, Nitish Garg, Jon Knight, and Tatsunori Hashimoto. Evaluating tool-augmented mathematical reasoning through instruction tuning. *arXiv preprint arXiv:2305.13535*, 2023.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.

Chuancai Li, Yuhao Wang, Tao Shen, Zhen Wang, Xiangyu Zhang, Weinan Zhang, Zheyan Liu, Tianye Ying, Yue Chen, Yuxuan Ding, Xinjun Peng, Yongdong Zhang, and Feng Wu. Thinking process distillation: Test-time thinking process distillation for few-shot prompting to enhance reasoning capability of large language models. *arXiv preprint arXiv:2312.05500*, 2023.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL `http://arxiv.org/abs/1707.06347`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Peiyi Wang, Liang Li, Zhihong Shao, Runxin Xu, Daya Dai, Yong Li, Deli Chen, Yuehua Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023a.

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

Jérémy Yang, Md Rizwan Gupta, Peter Goldman, David Brandfonbrener, Hancheng Wang, and Paul Christiano. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*, 2023.

Yifan Zhang, Jingqin Yang, Xinyi Wang, Yifan Gao, Heng Ji, and Avi Sil. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

# A Author Contributions

- **Abhinav Agarwal**: Dataset creation and curation, fine-tuning methodology, GRPO implementation and optimization
- **Carlo Baronio**: Fine-tuning approaches, GRPO training and theoretical analysis, system optimizations
- **Shubhra Mishra**: Multi-agent framework, PRM reranking, verification systems
- **Shree Reddy**: Multi-agent solution generation, preference modeling, evaluation methodology

# B Extended Experimental Details

This section contains additional details about experiments that were omitted from the main paper due to space constraints.

## B.1 Token Usage Analysis

Our token efficiency analysis demonstrated that models trained with our pipeline learned to adaptively allocate tokens based on problem difficulty. For simple problems, the model used an average of 2,184 tokens, while for complex problems requiring detailed reasoning, it used up to 7,912 tokens. This adaptive allocation resulted in an overall average of 5,048 tokens per problem.

## B.2 Tool-Integrated Reasoning

Tool-Integrated Reasoning (TIR) showed consistent improvements across model scales, with the most significant relative gains observed in smaller models. This suggests that external tools can partially compensate for limited model capacity, though our GRPO approach achieved superior results overall.