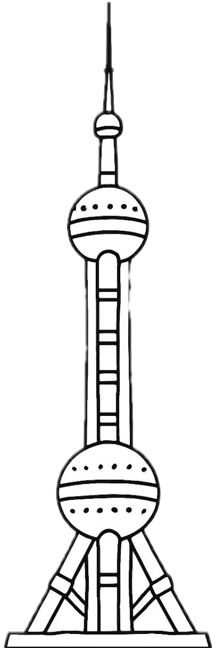
Towards a More Efficient Reasoning LLM:

AIMO2 Solution Summary and Introduction to Fast-Math Models



8 June 2025 @ Shanghai Kaggler 2025

Aillis Inc., Senior ML Engineer | Univ. of Tokyo, Researcher

Kaggle Grandmaster @analokamus

Hiroshi Yoshihara | 吉原 浩之

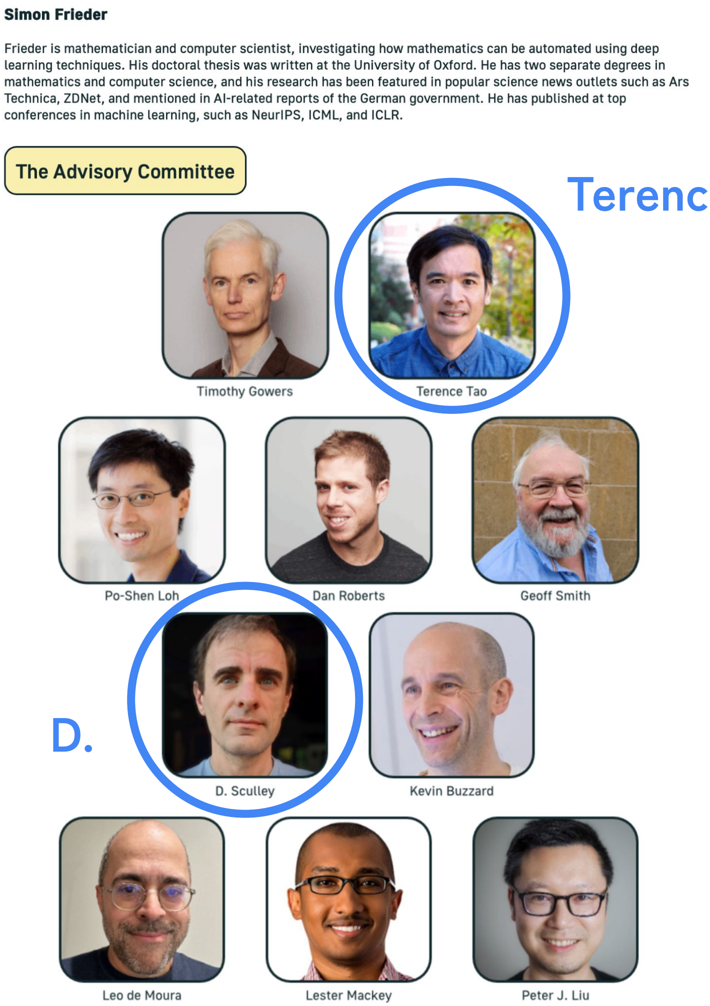
Agenda

1. AIMO2 Competition Overview
2. AIMO2 Top Solutions
3. Our Solution and Fast-Math Models

1.
AIMO2 Competition Overview

AIMO2 (AI Mathematical Olympiad Progress Prize 2)

- Objective: evaluate how well AI can acquire mathematical reasoning skills.
- Problem difficulty: AIMO1 < AIME (domestic competition) < AIMO2 < IMO (international competition)
- Each answer was a non-negative integer between 0 and 999.
- Intermediate reasoning or proofs were not evaluated. Only the accuracy of the final numerical answer was considered.



Simon Frieder

Frieder is mathematician and computer scientist, investigating how mathematics can be automated using deep learning techniques. His doctoral thesis was written at the University of Oxford. He has two separate degrees in mathematics and computer science, and his research has been featured in popular science news outlets such as Ars Technica, ZDNet, and mentioned in AI-related reports of the German government. He has published at top conferences in machine learning, such as NeurIPS, ICML, and ICLR.

The Advisory Committee

Timothy Gowers

Terence Tao

Po-Shen Loh

Dan Roberts

Geoff Smith

D. Sculley

Kevin Buzzard

Leo de Moura

Lester Mackey

Peter J. Liu

Terence Tao

D.

https://aimoprize.com/updates/2024-07-18-prize-manager-and-advisory-committee

FYI... Problem #1

Three airline companies operate flights from Dodola island. Each company has a different schedule of departures. The first company departs every 100 days, the second every 120 days and the third every 150 days. What is the greatest positive integer d for which it is true that there will be d consecutive days without a flight from Dodola island, regardless of the departure times of the various airlines?

Competition settings

- 10 example + 50 public LB + 50 private LB problems
- L4 x 4 Instance / 5 hours time limit
- No official training data
- One submission/day
- Only one question was visible at a time
    - No access to multiple questions simultaneously
    - Impossible to return to previous questions
- The order of questions was randomized for each submission (public LB only)

Challenge #1: Problem difficulty

- AIMO2 problems were remarkably more difficult than those in AIMO1
- In the beginning of AIMO2, open-source models on public LB:
    - NuminaMath-7B (AIMO1 1st place) ~2/50 (cf. 29/50 in AIMO1)
    - Qwen2.5-Math-72B-CoT ~5/50
    - Qwen2.5-Math-72B-TIR ~8/50
- Lack of deep (long) reasoning capability in LLMs

Emergence of long reasoners

- Some long reasoning models were released during this competition.
  - Nov 2024 Alibaba - QwQ-32B-Preview
  - Jan 2025 DeepSeek - DeepSeek-R1 and distilled models
- Long reasoners (w/o fine-tuning) on public LB:
  - QwQ-32B-Preview ~18/50
  - R1-Distilled-Qwen-14B ~27/50
- Long reasoning capability significantly raised the competition baseline.

How does long reasoning model works

- Reasoning model is trained to output a chain of thought (CoT) enclosed in <think> tags at the beginning of its response.
  - <think> CoT... </think> response...
- For math problems, the model is trained to output the final answer in LaTeX format using \boxed{}.
- The answer is often also output right before the closing </think> tag.
  - <think> CoT...\boxed{answer} </think> response...\boxed{answer}

Best public baseline notebook

- https://www.kaggle.com/code/octaviograu/lb-27-aimo-2-deepseek-r1-distill-qwen-7b-awq
- Model: R1-Distilled-Qwen-7B-AWQ served on vLLM
- Prompts: two different simple prompts mixed
- Token budget: 12000 or 8000, dynamic scheduling based on time left
- Answer processing: early-stop at </think> token, majority voting @ 32
- Public LB: 27/50 (results very unstable: score variance ~4)

Challenge #2: Reasoning capability

- Strategy #1: Enhancing CoT capability
    - e.g., Qihoo 360 - Light-R1 https://github.com/Qihoo360/Light-R1
- Strategy #2: Using TIR (Tool-Integrated Reasoning) capability
    - TIR: Ask model to output code instead of direct answer
    - TIR enables models to solve specific types of problems with brute force.
    - R1-distilled-Qwen models inherit TIR capability from its base model, Qwen2.5.

Qihoo 360 - Light-R1

- Starting from the non-reasoning Qwen2.5, SFT and RL led to surpassing the math performance of the R1-distilled model.
- Enhanced the existing R1-distilled model through SFT with a small amount of data.

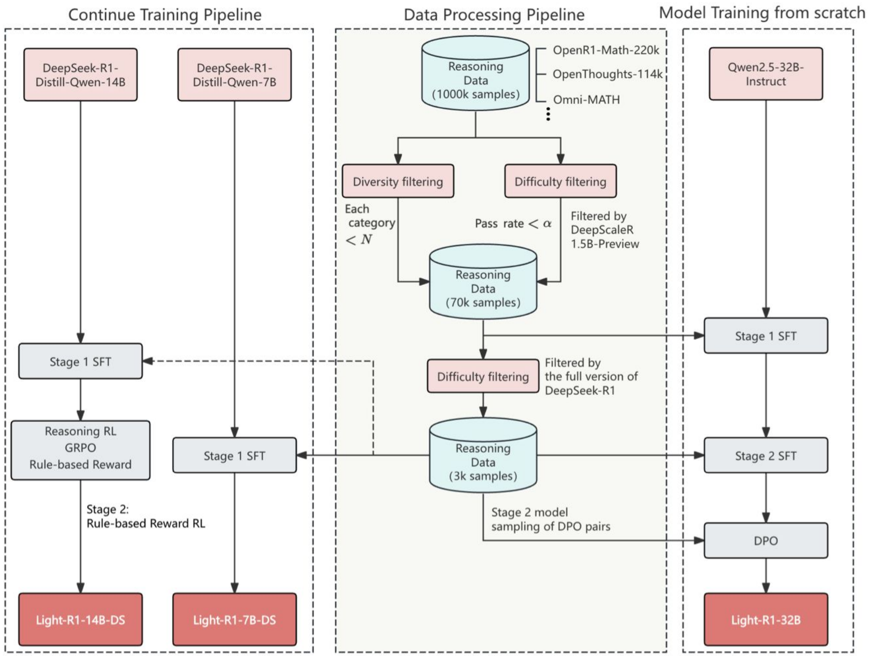
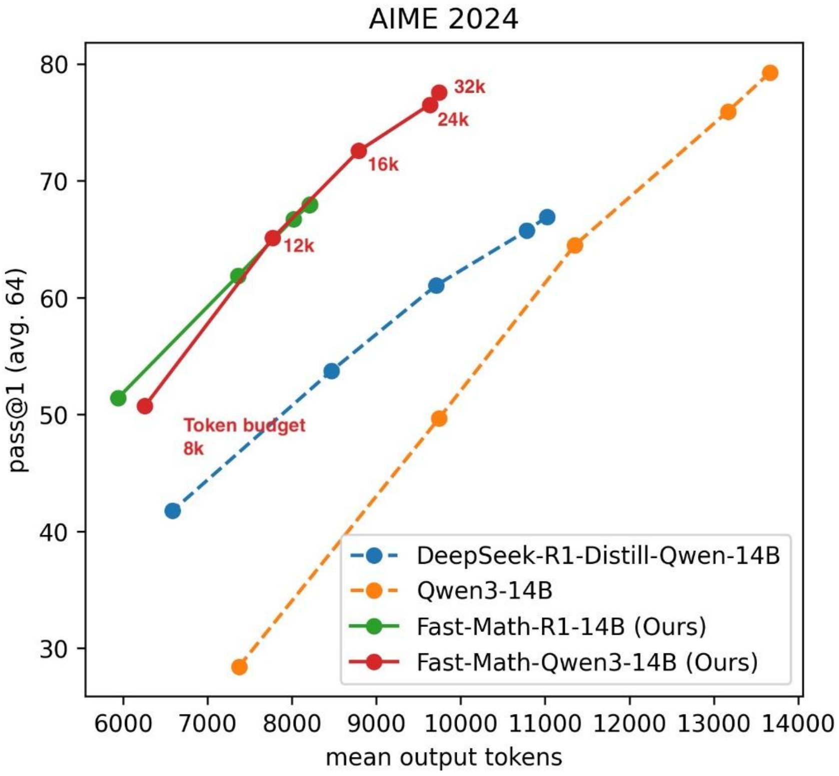

Continue Training Pipeline
Data Processing Pipeline
Model Training from scratch
DeepSeek-R1-Distill-Qwen-14B
DeepSeek-R1-Distill-Qwen-7B
Reasoning Data (1000k samples)
OpenR1-Math-220k
OpenThoughts-114k
Omni-MATH
Qwen2.5-32B-Instruct
Diversity filtering
Difficulty filtering
Each category $< N$
Pass rate $< \alpha$
Filtered by DeepScaleR 1.5B-Preview
Reasoning Data (70k samples)
Stage 1 SFT
Difficulty filtering
Filtered by the full version of DeepSeek-R1
Stage 1 SFT
Reasoning RL GRPO Rule-based Reward
Stage 1 SFT
Reasoning Data (3k samples)
Stage 2 SFT
Stage 2: Rule-based Reward RL
Stage 2 model sampling of DPO pairs
DPO
Light-R1-14B-DS
Light-R1-7B-DS
Light-R1-32B

Figure 2: Overview of training pipeline of Light-R1 series.

Challenge #3: Reasoning efficiency

- Longer reasoning generally produce better answers.
    - Reasoning efficiency: the number of tokens required to reach a answer
- Reported performance in papers and technical reports is typically based on generous token budgets (e.g., 32k tokens).
- In the AIMO2 submission environment, token budgets were practically limited to 8k–16k, depending on the model size.
- Many teams overlooked/underestimated this part.

Model performance under token budget restrictions



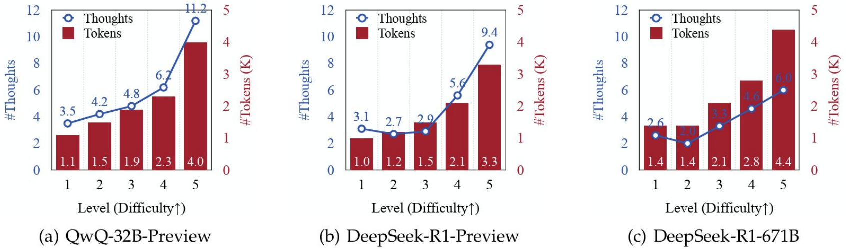
In AIME 2024 (with difficulty similar to AIMO2), a significant drop in accuracy occurs between 8k and 16k token budgets.

Strategies to deal with reasoning efficiency

- Strategy #3: Enhancing reasoning efficiency
  - e.g., On the Overthinking of o1-Like LLMs
    https://arxiv.org/pdf/2412.21187
- Strategy #4: Accelerating inference through hard/software optimization
- Strategy #5: Exploring quantization settings with minimal performance degradation

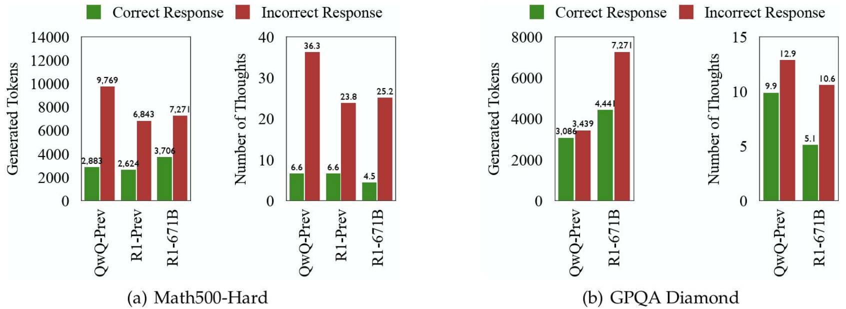
On the Overthinking of o1-Like LLMs

- A study on syntactic analysis of reasoning traces and the correlation between reasoning switches and accuracy.
- Improved performance on math tasks by applying constrained decoding that suppresses the probability of reasoning-switch prefixes in the n tokens following their appearance.



(a) QwQ-32B-Preview (b) DeepSeek-R1-Preview (c) DeepSeek-R1-671B

Figure 3: Average number of thoughts (“Thoughts”) and tokens (“Tokens”) in generated responses across different difficulty levels of the MATH500 test set.



(a) Math500-Hard (b) GPQA Diamond

Figure 4: O1-like LLMs switch thinking more frequently on incorrect responses, thus expend more tokens without contributing to accuracy.

# TODO:

- Strategy #1: Enhancing CoT capability
- Strategy #2: Using TIR capability
- Strategy #3: Enhancing reasoning efficiency
- Strategy #4: Accelerating inference
- Strategy #5: Exploring quantization settings

2.
AIMO2 Top Solutions

Team imagination-research (2nd place)

- Details: https://www.kaggle.com/competitions/ai-mathematical-olympiad-progress-prize-2/discussion/572948
- Members:
  - Yichen You: Tsinghua University
  - Xuefei Ning: Tsinghua University, project leader
  - Zinan Lin: Microsoft Research
- Public LB 34/50 (1st) - Private LB 31/50 (2nd)

Solution overview: imagination-research

- Part I: Reasoning-Oriented Training
    - Strategy #1 and #2: enhancing CoT / using TIR
- Part II: Efficiency Optimization
    - Strategy #4 and #5: accelerating inference / exploring quantization
- Part III: Inference-Time Strategies
    - Quite elaborate inference pipeline, but I will omit this part to focus on the model itself.

Reasoning-Oriented Training: First stage

- SFT
- dataset: Light-R1 second stage + LIMO (https://github.com/GAIR-NLP/LIMO)
- R1-distilled-Qwen-14B / 8 epochs
- The accuracy improves but the output length also improves significantly.

Second stage: Direct Preference Optimization (DPO)

- DPO reframes preference learning as a binary classification task, where the model is trained to assign higher likelihood to the preferred response over the less preferred one, based on comparisons.
- Team imagination-research created DPO training pairs based on three criteria: answer correctness, response length, and pairwise similarity.

Efficiency Optimization

- Using lmdeploy as the LLM inference framework, compared with vllm, can provide higher throughput and shorter model initialization time.
- Using 4-bit AWQ weight / 8-bit KV cache quantization (W4KV8) enabled ~25% faster inference compared to W4KV16 quantization with no accuracy degradation.

Team NemoSkills (1st place)

- Details: https://www.kaggle.com/competitions/ai-mathematical-olympiad-progress-prize-2/discussion/574765
- NVIDIA team: Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, Igor Gitman
- Public LB 32/50 (2nd) - Private LB 34/50 (1st)
- *"The training lasted 48 hours on 512 H100 (yes, 512!)"*

Solution overview: NemoSkills

- Model Training
  - Strategy #1 and #2: enhancing CoT / TIR
- Inference Optimization
  - Strategy #4 and #5: accelerating inference / exploring quantization
  - This part handles fancy backend inference using TensorRT, but I will skip the details here.

Model Training

- Created high-quality 2.2M math CoT dataset using DeepSeek-R1.
- Created high-quality 15k math TIR dataset.
- First stage: Qwen2.5-14B / SFT / 8 epochs / CoT dataset
- Second stage: first-stage model / SFT / 400 steps / TIR dataset
- Final model is a merged model: CoT * 0.3 + TIR * 0.7
- 512 x H100 x 48 hrs

Inference Optimization

- ReDrafter, a speculative decoding technique implemented in TensorRT-LLM was used to. ReDrafter head was trained on a random subset of problems from OpenMathReasoning-1 dataset.
- Inference speed on TensorRT-LLM: bf16 < int8 ~ fp8 < int4 < fp8 + Redrafter
- Accuracy: int4 < all others

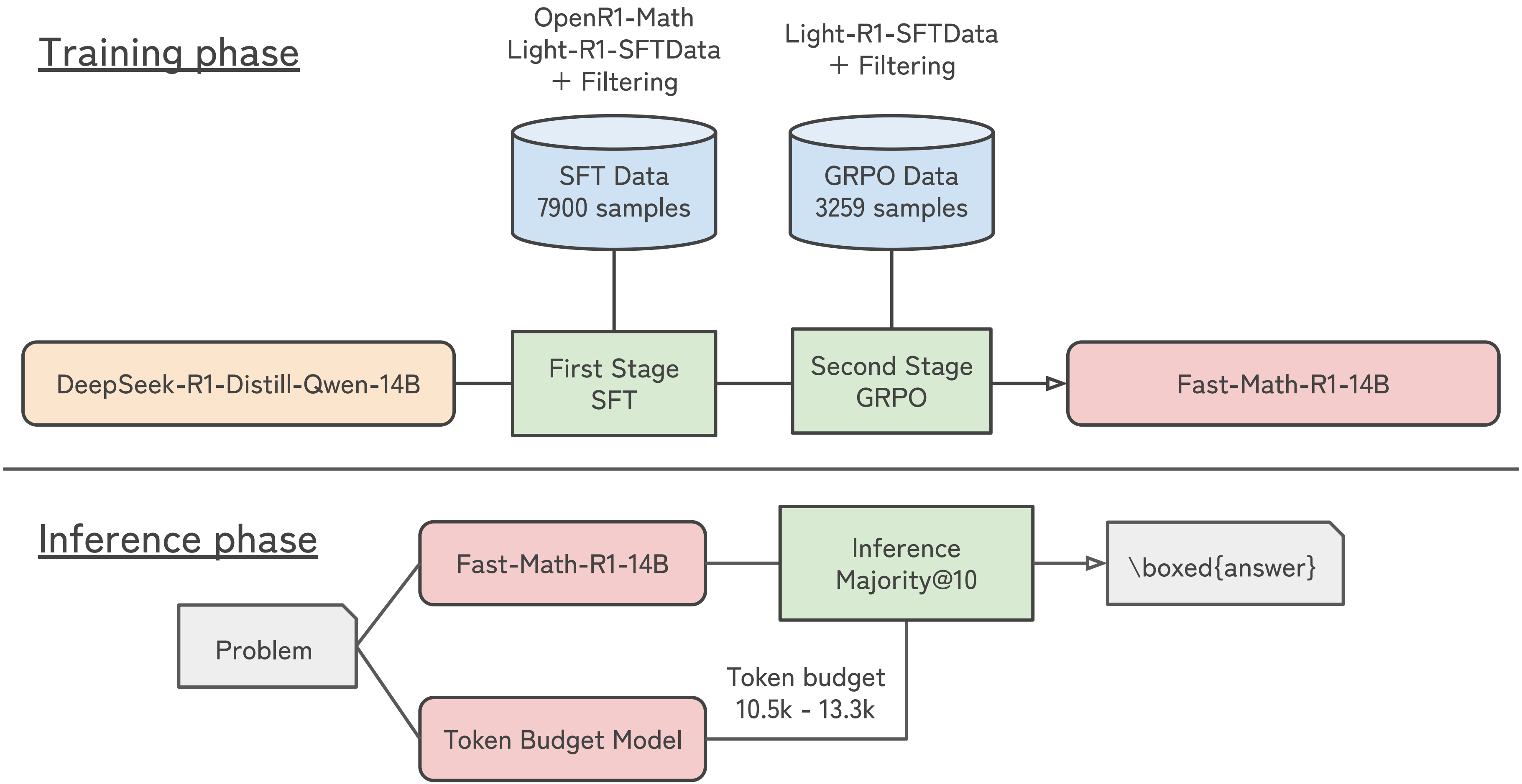
3.
Our Solution and Introduction to Fast-Math models

Team Fast-Math-R1-14B (天才受験生と呼ばれたものたち)

- Details: https://www.kaggle.com/competitions/ai-mathematical-olympiad-progress-prize-2/discussion/571252
- Members:
  - Hiroshi Yoshihara: Aillis Inc., The University of Tokyo
  - Yuichi Inoue: Sakana AI Co, Ltd.
  - Taiki Yamaguchi: Rist Inc.
- Public LB 29/50 (7th) - Private LB 28/50 (9th)

Solution overview: Fast-Math-R1-14B

- First stage: intensive SFT using a high-difficulty dataset
    - Strategy #1: enhancing CoT
- Second stage: GRPO for more efficient reasoning
    - Strategy #3: enhancing reasoning efficiency
- Inference time scheduling



Training phase

OpenR1-Math
Light-R1-SFTData
+ Filtering

Light-R1-SFTData
+ Filtering

SFT Data
7900 samples

GRPO Data
3259 samples

DeepSeek-R1-Distill-Qwen-14B

First Stage
SFT

Second Stage
GRPO

Fast-Math-R1-14B

Inference phase

Problem

Fast-Math-R1-14B

Token Budget Model

Inference
Majority@10

Token budget
10.5k - 13.3k

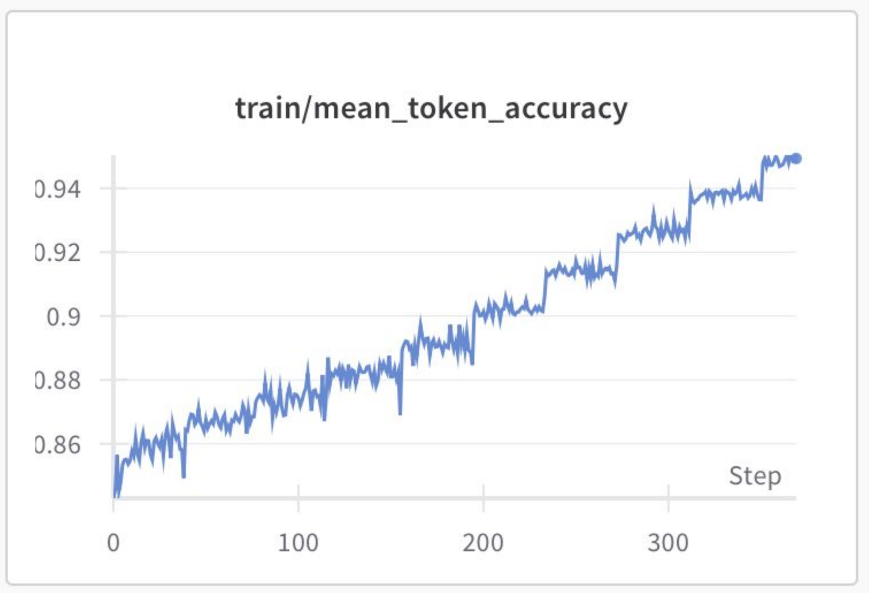
\boxed{answer}

Dataset for the first stage SFT

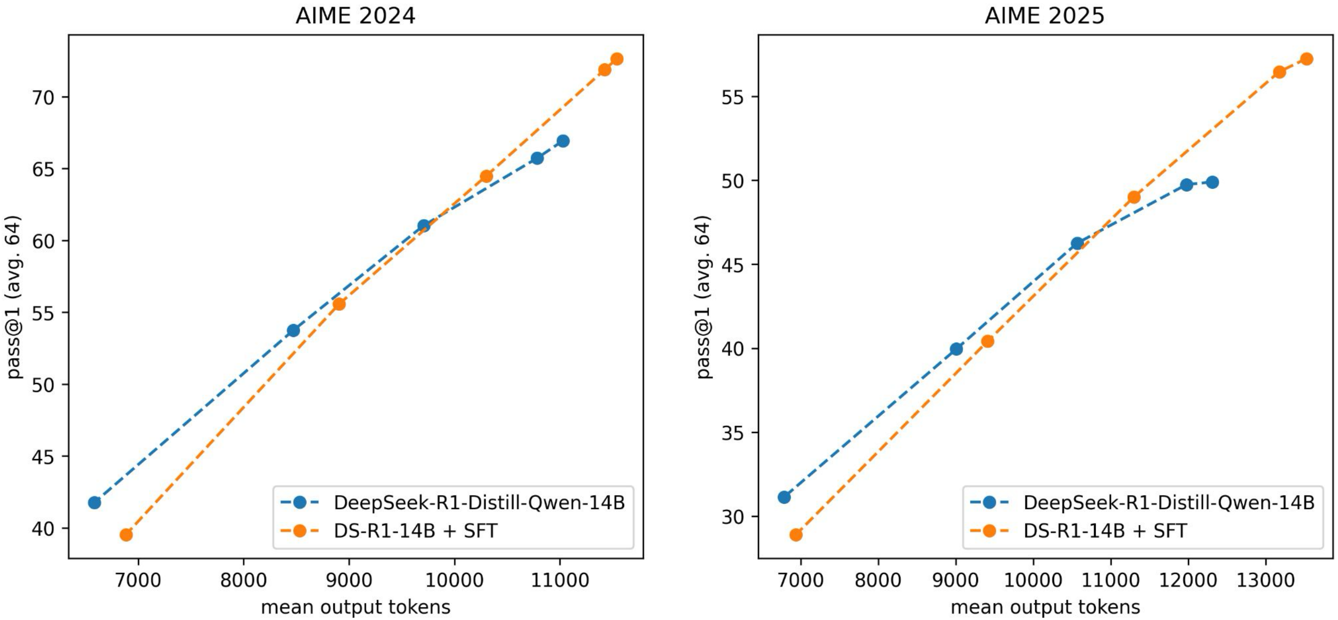
- We sampled high-difficulty (low accuracy with R1) problems from the OpenR1-Math and Light-R1 second-stage datasets, and generated 7900 problem - R1 shortest correct trace pairs.
- Quality of the dataset (difficulty, answer length, and diversity) was crucial in SFT.
    - SFT on easy problems resulted in a model that quickly produces incorrect answers for difficult questions.

First stage settings

- SFTTrainer from trl was used
- 7900 problem-trace pairs
- Full-parameter tuning
- 10 epochs
- Training time: approx. 10 hours (8 x H200)



Results of first stage (SFT)

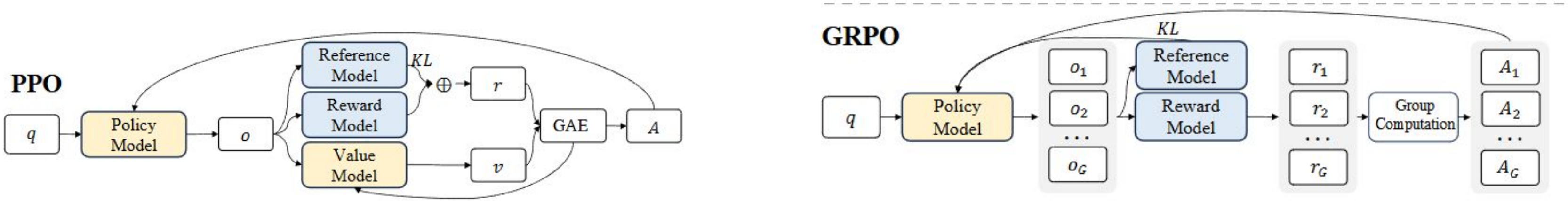


Peak performance improved at token budgets > 24k, but...

Interpretation of the first stage results

- On public LB, R1-distilled-Qwen 14B ~25/50 vs. SFT ~24/50 (worse!)
- SFT does improve model performance when token budget is unlimited.
- Under the constraints of this competition, it is necessary to improve efficiency while maintaining model accuracy.
- R1's CoT is quite verbose.

Group Relative Policy Optimization (GRPO)



- New RL algorithm used in DeepSeek R1 https://arxiv.org/abs/2501.12948
- Unlike conventional PPO (Proximal...), GRPO removes value model and instead uses Monte Carlo estimation of expected rewards (by sampling multiple responses to the same question), reducing computation and improving training stability.
- Well-suited for tasks like math problems, where rewards are easy to define.

Second stage: GRPO for more efficient reasoning

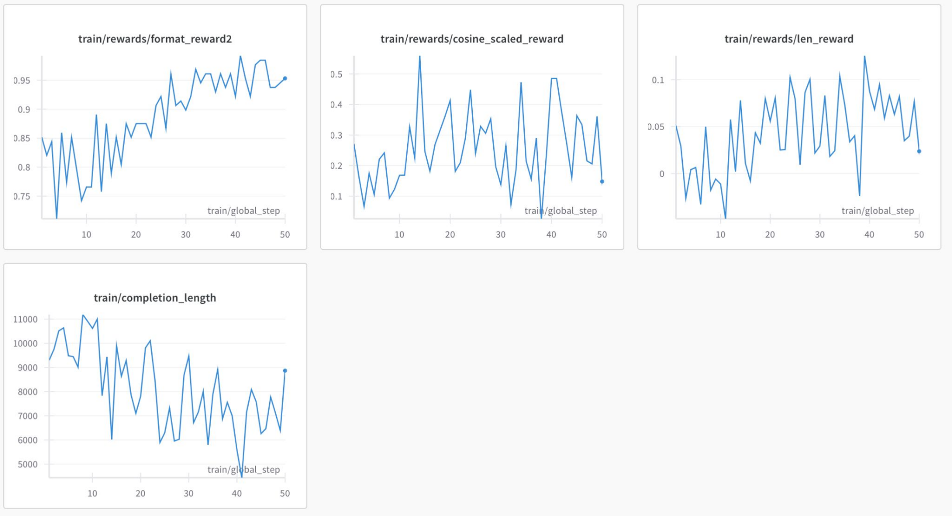
- Problem–answer pairs were extracted from the Light-R1 dataset and used for GRPO training.
- Three types of reward functions were used:
  - Format reward: in order to save output tokens, we forced the model to give an answer in the end of reasoning block before </think> by rewarding the pattern `r"^.*?oxed{(.*?)}.*?</think>.*?$"`. Generation is stopped at </think> during inference.

Second stage: GRPO for more efficient reasoning

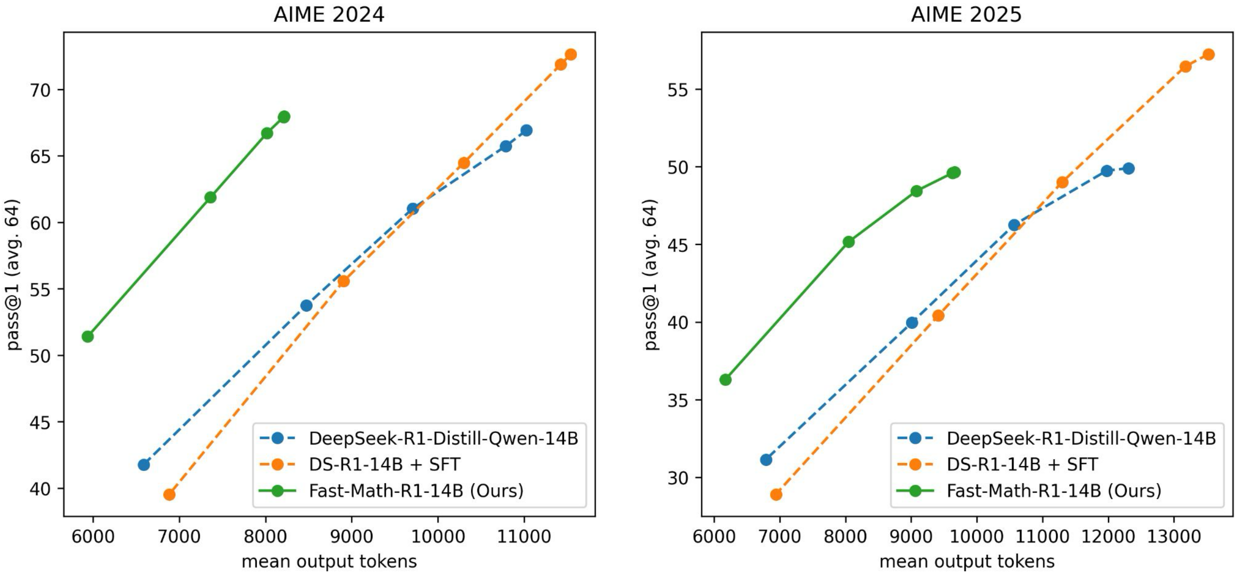
- Three types of reward functions were used:
  - Cosine reward: compared to a normal accuracy-based reward, cosine reward applies a continuous penalty to longer correct reasoning traces and shorter incorrect ones.
  - Length reward: length-based rewards to discourage overthinking and promote token efficiency. https://arxiv.org/abs/2501.12599
  - Total reward = format reward + cosine reward + length reward

Second stage settings

- GRPOTrainer from trl was used
- 3259 problem-answer pairs
- Full-parameter tuning
- 50 steps (~0.25 epoch)
- Training time: approx. 10 hours (8 x H200)

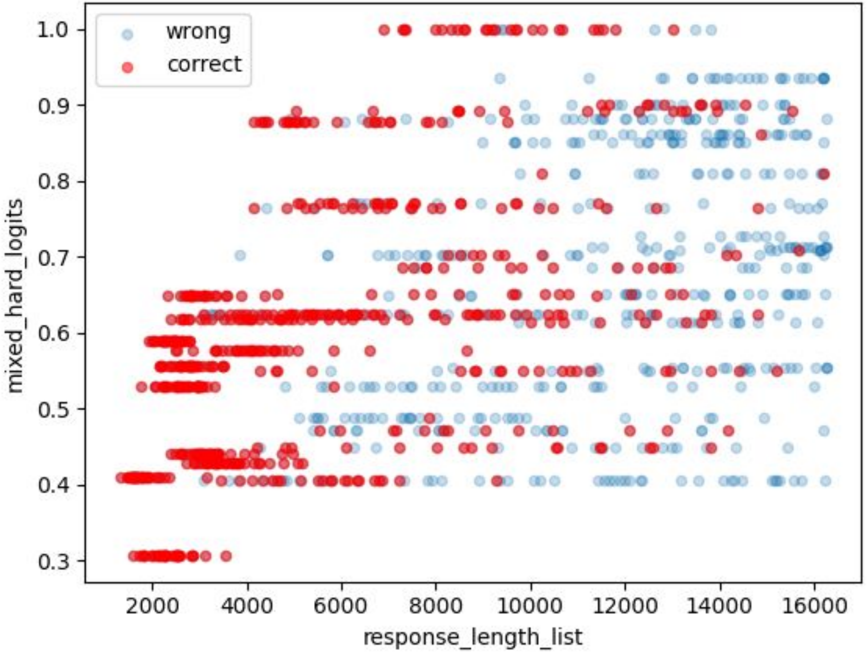


Results of second stage (GRPO)



The new model outperforms the original in both peak performance and inference efficiency, achieving the same accuracy with an average of 30% faster inference.

Inference time scheduling

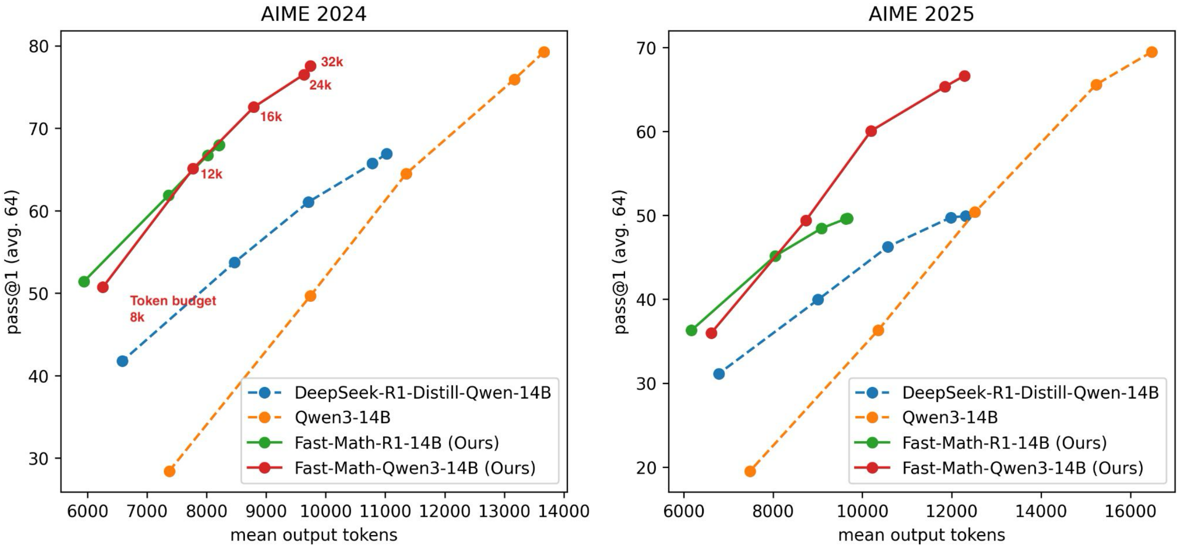


We trained a ModernBERT model to predict the difficulty of problem, defined by the number of tokens required to reach a correct answer. We used this model to adjust inference time for each problem.

Final results and key takeaways

- GRPO pushed our public score from 25 to 29, and stabilized the score.
- The score gap between us and top teams is likely the use of TIR.
- SFT is good at adding new knowledge to the model and improve the peak performance.
- GRPO is a powerful RL method for aligning how a model leverages its knowledge to specific goals, and is especially effective in tasks like math where rewards are easy to define.

Fast-Math model family



When we applied our GRPO recipe from the AIMO2 competition to other models and benchmark datasets, it consistently yielded strong results.

Fast-Math family is fully open-source

- We have released all datasets, code, and model weights used to train the Fast-Math models.
    - Model weights (DeepSeek Qwen 2.5, NVIDIA OpenMath, Qwen3 variants) and datasets (Huggingface)
    - Code (Github)
- A paper detailing the technical methodology and ablation studies is currently in preparation.

Thank you for listening :)