Universiteit Leiden
The Netherlands

Opleiding Informatica

Reinforcement Learning for Training Small LLMs by Distillation

Xu Li

Supervisor:
Aske Plaat
Second Reader:
Niki van Stein

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Abstract

This study aims to investigate whether reinforcement learning (RL) can effectively improve the performance of knowledge distilled small language models in mathematical reasoning field under constrained conditions. Our experiment used Qwen2.5-32B-Instruct as the student model, which is distilled by two teacher models, GPT-4.1 and Claude Sonnet 4, and then followed by GRPO fine-tuning with a small sample size. Our findings reveal that under constrained conditions, the performance improvements from distillation are limited, and subsequent RL fine-tuning not only yields unstable results but may even lead to performance degradation. We also found that to obtain a better distillation result the alignment between teacher and student capabilities was more critical than the teacher's absolute strength.

Contents

Chapter 1

Introduction

Large Language Models (LLMs) [NKQ+23] are neural networks with hundreds of billions of parameters pre-trained on vast amounts of data. They possess good general language capabilities and can serve as Foundation Models for various downstream tasks [IBM23]. In recent years, research on LLMs has made significant progress in the field of artificial intelligence (AI), demonstrating remarkable performance in tasks such as natural language understanding, content generation, and complex reasoning [ZZL+23]. As a result, LLMs have been widely applied both in academia and industry [ZZL+23].

However, the performance of these LLMs relies on their large parameter scale and computational requirements [KMH+20]. The training and deployment of LLMs usually not only require a large amount of computing power, but also entail significant energy consumption and economic costs, which limits their application in resource-constrained environments for individuals and small corporations [BGMMS21]. Therefore, how to migrate and encapsulate the capabilities of those LLMs into smaller, more efficient models while maintaining similar performance level has become one of the key study directions related to the field of LLMs [ZLL+24b].

As one of the mainstream strategies to address this challenge, knowledge distillation is an effective technique for model compression and knowledge transfer. Its core idea is to train a smaller "student" model to learn from and reproduce the capabilities of a more powerful "teacher" model [IBM23]. Additionally, some research is exploring the application of RL (reinforcement learning) to further optimize the effect of distillation [TBL+23]. RL [SB18] is a computational framework that learns from 'trial and error.' During the RL training process, an agent (or model) interacts with the environment and receives rewards or penalties based on its actions. Through this feedback mechanism, the agent can continuously optimize its decision-making policy to achieve specific goals. In the domain of LLMs, this technique typically manifests as Reinforcement Learning from Human or AI Feedback (RLHF/RLAIF) [OWJ+22], for which Proximal Policy Optimization (PPO) [SWD+17] is the most classic and widely used algorithm. In recent years, to enhance training efficiency and stability, the academic community has also developed several innovative variants, such as Direct Preference Optimization (DPO) [RSM+23] and Group-wise Reward Policy Optimization (GRPO) [PJ25].

1.1 Problem Statement

Specifically, this paper focuses on the following research goals:

Problem statement: Can reinforcement learning improve distillation for trainning high-performance small LLMs?
This problem statement can be divided into the following three research questions.

- **RQ1:** Can small models trained via distillation techniques with limited data size outperform large models in mathematical reasoning tasks?
- **RQ2:** Is GPT or Claude a better teacher model for distillation?
- **RQ3:** Can subsequently fine-tuning a distilled student model with RL further improve the performance?

1.2 Method

To investigate the above questions, this study will first construct a sample question bank based on three open-source mathematics datasets: OpenAI Math [Ope21], NuminaMath [AI-24], and JEEBench [DAI24]. Subsequently, GPT-4.1 [Ope25] and Claude Sonnet 4 [Ant25] will be selected as teacher models to separately generate Question-Reasoning Trace-Answer (Q-R-A) triplets from the original question bank, and high-quality data will be screened for distillation training. The study will then select Qwen2.5-32B-Instruct [Qwe25] as the baseline student model for distillation and attempt further fine-tuning using RL (GRPO). The performance of the final models will be evaluated using two mathematical benchmarks: AIME 2024 [Hug24b] and GSM8K [Hug21].

1.3 Overview

In the remainder of this thesis, in Chaper 2, we will discuss related work. In Chapter 3, we will introduce our experimental setup in details. Following this, in Chapter 4, we will present all the experiment results, including training metrics and the final benchmark scores. In Chapter 5, we will analyze and interpret our findings, and offer potential explanations for them. Finally, in Chapter 6, we will conclude this study by summarizing our answers to the research questions and problem statement, discussing the study's limitations, and suggesting possible further research.
To facilitate further research, our code and models are made publicly available at `https://github.com/Alizabethli/llm-distillation-research` and `https://huggingface.co/Alizabethli`.

Chapter 2

Related Work

In this Chapter, we will discuss the related work regarding each research question, on which the scope of this study is inspired and the designed based.

2.1 Distillation Techniques

As stated in the previous chapter, the costs associated with training and deployment of LLM can be prohibitively high. Consequently, knowledge distillation has emerged as a promising approach to mitigate these substantial resource demands. The DeepSeek team [Gea25] used a large model optimised by reinforcement learning, DeepSeek-R1, as the distillation teacher model to train multiple smaller student models. In benchmarks of multiple domains, including mathematical reasoning, the performance of the student models surpassed that of GPT-4o-0513 [Ope24a]. Another study [MYS+25] also noted that using a very small amount (only 1,000) of high-quality Q-R-A triplet samples generated by Gemini Thinking Experimental as the teacher model could train smaller models that perform excellently on benchmarks such as MATH [HBK+21] and GSM8K. Except for the two studies mentioned above, there are also other studies explored the use of distillation to transfer the reasoning capacity from large models into smaller ones.

Zhu et al. [ZLL+24a] introduced a method called Equation-of-Thought Distillation (EoTD), which encodes the reasoning process into an equation-based format. Building on this, they proposed the Ensemble Thoughts Distillation framework by combining multiple types of reasoning traces, such as Chain-of-Thought (CoT) [WWS+22], Program-of-Thought (PoT) [CJLW22], and Equation-of-Thought (Eot) [ZLL+24a], to construct a refined dataset for fine-tuning smaller models. This approach aims to enhance the generalization capabilities of these models. Impressively, their 250M parameter student model achieved performance on mathematical reasoning benchmarks like GSM8K that was comparable to, or even surpassed, mid-sized open-source models like Vicuna-1.3B [LMS23] and CodeLLaMA [RGG+23]. However, on the other hand, it still lagged behind larger commercial models such as GPT-4 [Ope23] or Claude-2 [Ant23] due to its small parameter scale.

Shridhar et al. [SSS23] proposed an alternative strategy called Socratic Chain-of-Thought (Socratic CoT). This method decomposes complex math problems into subproblems and separately trains two small models: a problem decomposer and a subproblem solver. This modular approach mitigates the challenge that small models face in long-chain reasoning tasks. Tested on benchmarks such as GSM8K, the research demonstrated that even a model only having the size of GPT-2 Large could

outperform larger models like GPT-3 6B. This highlights the effectiveness of structured, modular distillation. However, they also found out that the dual-model setup increased training complexity, and its efficacy was limited on harder tasks.
These studies suggest that through well designed distilllation, small models can achieve impressive math and reasoning capacity even with limited training resources. In some tasks, they may even outperform models whose parameter scales are several times larger.

2.2 Effect Of Different Teacher Models

Whether small models trained by distillation can achieve good performance in math and reasoning related tasks can depend on the source and quality of the training data. The effect is especially relevant in resource-limited scenarios, where the choice of teacher plays a critical role in the final performance of the student models.
Bansal et al. [BHA+24] conducted a systematic study on knowledge distillation. It analyzed data generated by teacher models of varying scales and capabilities, including models from the Gemma 2 [Hug24a] and Gemini 1.5 [GT24] families under a fixed compute budget. Their findings revealed a critical trade-off: while data from larger, higher-performing teacher models exhibited greater accuracy, data from smaller, less capable models offered superior coverage and diversity. Surprisingly, student models trained on data generated by the smaller teachers ultimately outperformed those trained by larger teachers on several reasoning tasks. The study hypothesizes that this effect stems from improved learning efficiency. As the data from less powerful teachers is closer to the student model's own generative distribution, it provides a more effective learning signal for the students to improve themselves.
Koo et al. [KHK+24] investigated the impact of different teacher models on knowledge distillation from the generation perspective. Their study indicates that if a teacher model's output token distribution diverges significantly from the student's, directly using the teacher's outputs for supervised learning can cause a distributional mismatch. This mismatch can negatively affect both training stability and the final student model's performance. To address this problem, they proposed a strategy named SWITCH, which selectively uses teacher's outputs as the training target only when the distributional gap between the teacher and student models is substantial. This selective intervention approach demonstrated objective improvements in the experiments. This study highlights the importance of maintaining distributional alignment between teacher and student models in the distillation process.
Another related study by Li et al. [LYX+25] explored a phenomenon known as the "Small Model Learnability Gap." Their research indicates that complex reasoning chains, especially long CoTs generated by powerful teacher models are not always optimal for training small models ($\leq$3B parameters). Although these CoTs may be more accurate, their complexity often exceeds the learning capacity of small models, thereby undermining both alignment fidelity and generalization robustness. To address this limitation, the authors proposed a Mix Distillation strategy. This approach involves blending training data by combining both long and short CoTs (Mix-Long) or by sourcing data from both large and small teacher models (Mix-Large). This strategy can better align the complexity of training data with the capacity of the student model. The results of this study showed that models trained via Mix Distillation strategy significantly outperformed those trained with traditional single-teacher distillation strategy on reasoning benchmarks such as MATH.

In summary, existing research indicates that the efficacy of knowledge distillation is critically dependent on the selection of the teacher model. This topic presents valuable opportunities for further exploration.

2.3 Reinforcement Learning

RL has been widely used in the development of LLMs [KWBH24]. Unlike traditional supervised fine-tuning (SFT) [OWJ+22], which usually trains a model on static input-output examples, RL introduces a dynamic reward mechanism. This allows the model to be optimized for specific objectives, including complex tasks such as high-quality reasoning and mathematical problem solving. The DeepSeek team [Gea25] also highlighted the potential of applying RL (GRPO) as a subsequent fine-tuning step for distilled small models to further improve their performance, which inspired the scope design of this study.

Yin et al. [YZW+25] identified a learning capacity limitation as a bottleneck in the distillation process, particularly when small models are trained on long and complex CoTs. They argue that such models often struggle to fully capture these complex reasoning distributions. Hence, they proposed a hybrid strategy that combines tree-structured reasoning (using Monte Carlo Tree Search (MCTS) [BPW+12] to build tree-based CoTs), with RL techniques like Fine-grained DPO and joint training objectives. This approach is designed to mitigate model overfitting on long reasoning chains. Study results demonstrated that small models trained with this method significantly outperformed standard SFT baselines on benchmarks such as GSM8K.

Zhang et al. [ZWF+25] introduced a strategy named "distill data + rewards", which integrates SFT with RL. In this two-stage process, small models are first trained on teacher-generated data via SFT. Subsequently, an automatic scoring system is used to evaluate the quality of the model's reasoning outputs. These scores are considered as reward signals for a second stage of RL-based fine-tuning. This study illustrated that models trained with this combined approach achieved notable performance improvements on benchmarks such as GSM8K. Remarkably, the student models even surpassed the performance of the original teacher model in some cases.

These studies demonstrate that a subsequent phase of RL following distillation can be an effective strategy for enhancing the performance of small models on mathematical and reasoning tasks.

Chapter 3

Experimental Setup

In this Chaper, we will discuss our experimental setup. First, we will discuss the constrution of the raw dataset. Second, we will discuss how we select and generate our training datasets. Third, we will discuss the detailed training setup of distillation and subsequent reinforcement learning (GRPO). Finally, we will discuss the approach of performance evaluation.

3.1 Construction of the raw dataset

We chose three publicly available mathematical datasets (NuminaMath [AI-24], JEEBench [DAI24], and OpenAIMath [Ope21]) as the original datasets for this study. Python scripts were utilised to standardise the data structure of the raw data from the three sources, ensuring structure consistency in the final dataset for downstream experiments. After the standardisation process, all data was converted to JSON Lines (JSONL) format, with each line containing a single sample. Each sample includes a question, a final answer, and meta information (source). The specific data structure conversion is shown in Table 3.1. The detailed processing of the three datasets is as follows:

1. **NuminaMath Dataset:** This dataset was manually downloaded from Hugging Face. The raw data was provided in five Parquet files, containing a total of 859,494 samples. In this dataset, the reasoning steps and final answers are stored in the same field named `solution` in LaTeX format. To structurally separate them, we used script to automatically identify the last `\boxed{...}` command in the text and extract the contents inside as `final_answer`. All the preceding text left was treated as `reasoning`. Subsequently, we employed stratified sampling to select 3,000 samples, ensuring a balanced distribution across the dataset's nine sub-sources. The proportion of each topic in the sampled dataset is shown in Figure 3.1.

2. **JEEBench dataset:** This dataset was manually downloaded from its official GitHub repository. It is a single JSON file containing 515 samples. Since this study mainly focuses on the model's mathematical reasoning ability, only 236 questions labeled as subject 'math' were selected and retained. The `question` and `gold` (standard answer) attributes were mapped into standardized data structure. Other valuable attributes, such as `subject`, `type` (e.g., MCQ, Numeric), and `description`, were retained as metadata in case of further refined processing. The type distribution can be seen in Figure 3.2.

3. **OpenAIMath Dataset:** This dataset was manually downloaded from the official OpenAI GitHub repository. It contains 7,473 elementary-level math problems in JSONL format. Similar to NuminaMath, the `answer` attribute in this dataset also combines reasoning with final answer. Hence, we split it into two attributes, `reasoning` and `answer`, by recognizing the pattern `\n####` as explicit separator. After splitting, 1,000 samples were randomly selected.

After performing preliminary standardization and sampling on the three data sources, we integrated them into the final experimental question bank. This process included several quality control steps. First, we unified the structure of all data, ensuring that each entry contained only the `question`, `final_answer`, and simplified `metadata` attributes. Second, we removed all the invalid entries having empty fields and performed strict deduplication regardless of letter case. Finally, we also assigned a unique sequence ID `math-q-XXXXX` to each valid data entry for data tracing. After all these processing steps, a final non-redundant math question bank containing 4,125 entries was created.

To construct a moderately sized and representative training dataset, we then performed another stratified sampling on the remaining 4,125 questions by data sources, ultimately selecting 3,000 samples for the teacher model to generate Q-R-A triplets. The source distribution of the selected dataset was as follows: NuminaMath (2,104 questions), OpenAIMath (724 questions), and JEEBench (172 questions). The distribution is illustrated in Figure 3.3.

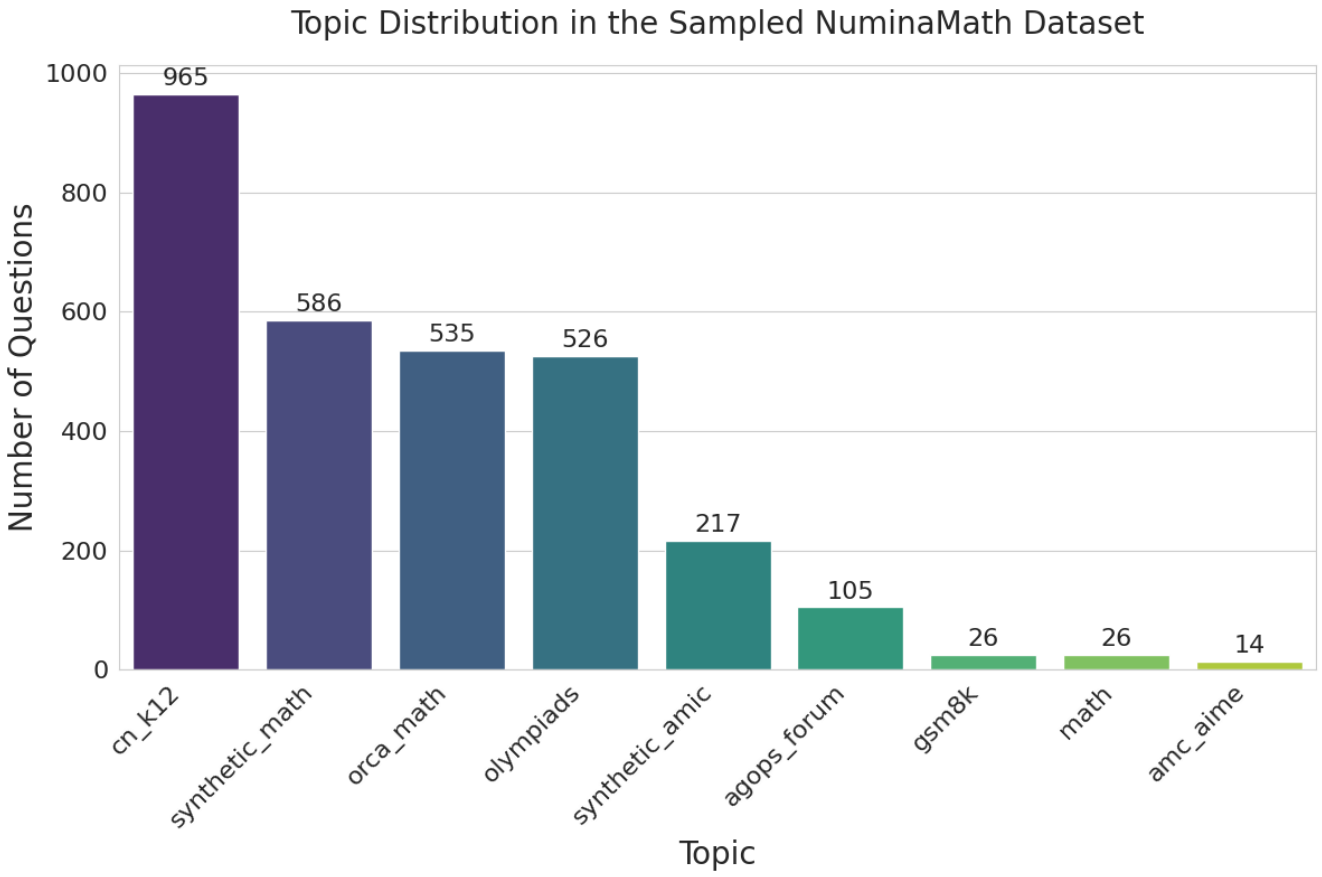

Figure 3.1: Topic distribution in the sampled NuminaMath dataset.

Table 3.1: Original Data Structures and Standardization Conversion

Category	Data Source	Data Structure Example

After Processing (Target Format)	All Sources	**Description:** Standardized into a uniform structure containing question, reasoning, answer, and metadata.
		`json {`
		`  "question": "...",`
		`  "final_answer": "...",`
		`  "metadata": {`
		`  "source": "OpenAIMath", ...`
		`  }`
		`}`

Before Processing (Original Format)	NuminaMath	**Description:** Parquet file format. The `solution` field mixes reasoning and the final answer in LaTeX format.
		`json {`
		`  "source": "...",`
		`  "problem": "...",`
		`  "solution": "... \boxed{...}"`
		`  "message.list": "...",`
		`}`

	JEEBench	**Description:** JSON file format. Contains rich metadata fields.
		`json {`
		`  description": "JEE Adv 2023 Paper 2", ...`
		`  "index": "...",`
		`  "subject": "math", ...`
		`  "type": "MCQ", ...`
		`  "question": "...",`
		`  "gold": "..." (answer),`
		`}`

	OpenAIMath	**Description:** JSONL file format. The `answer` field uses the `\n####` delimiter to separate reasoning and the final answer.
		`json {`
		`  "question": "...",`
		`  ... \n#### ... (answer)"`
		`}`

3.2 Distilled training data generation

In this stage, we chose two commercial LLMs as our teacher models to construct a high-quality Q-R-A triplet dataset based on the 3,000 math problems collected in the previous stage. For generating high-quality reasoning traces, we employed two leading closed-source models as teachers: OpenAI's GPT-4.1 and Anthropic's Claude Sonnet 4, both distinguished by their advanced reasoning abilities. These two teachers independently answered the 3,000 questions to generate two parallel Q-R-A triplets datasets for the following distillation.

We designed and adopted an automated process to generate triplets by querying the two teacher models via API. In the practice, we chose to use the OpenRouter API platform [Ope24b] to query the teacher models. To ensure consistency in the generated content and reduce randomness, the following generation parameters were fixed as follows:

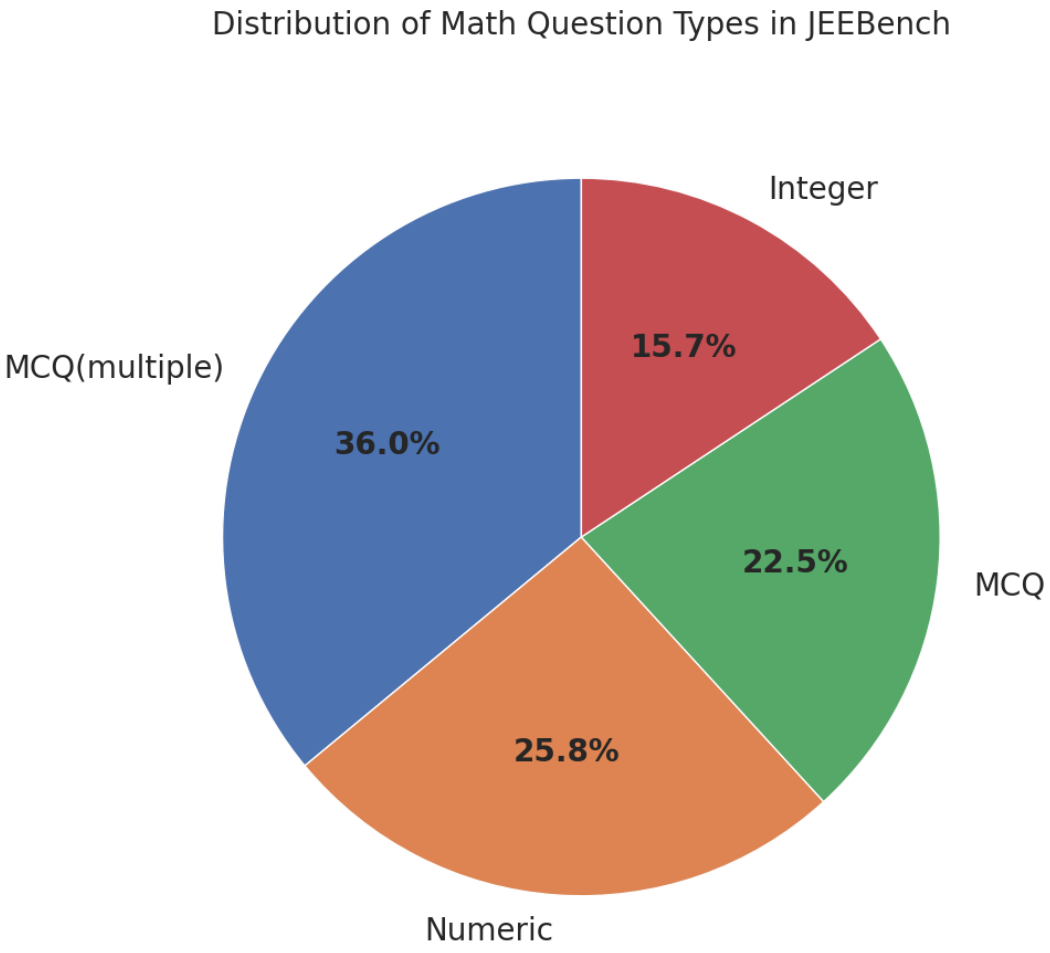

Figure 3.2: Type distribution in the JEEBench dataset.

- **Temperature:** 0.2, to produce more deterministic and factual outputs.
- **Max Tokens:** 8192, to ensure sufficient space to accommodate the complete reasoning trace for complex problems.
- **Timeout:** 360 seconds, to allow ample time for the model to process complex reasoning.

Additionally, in order to guide the teacher models to generate structured, good-quality reasoning trace, we designed a unified prompt as shown in Box 1.

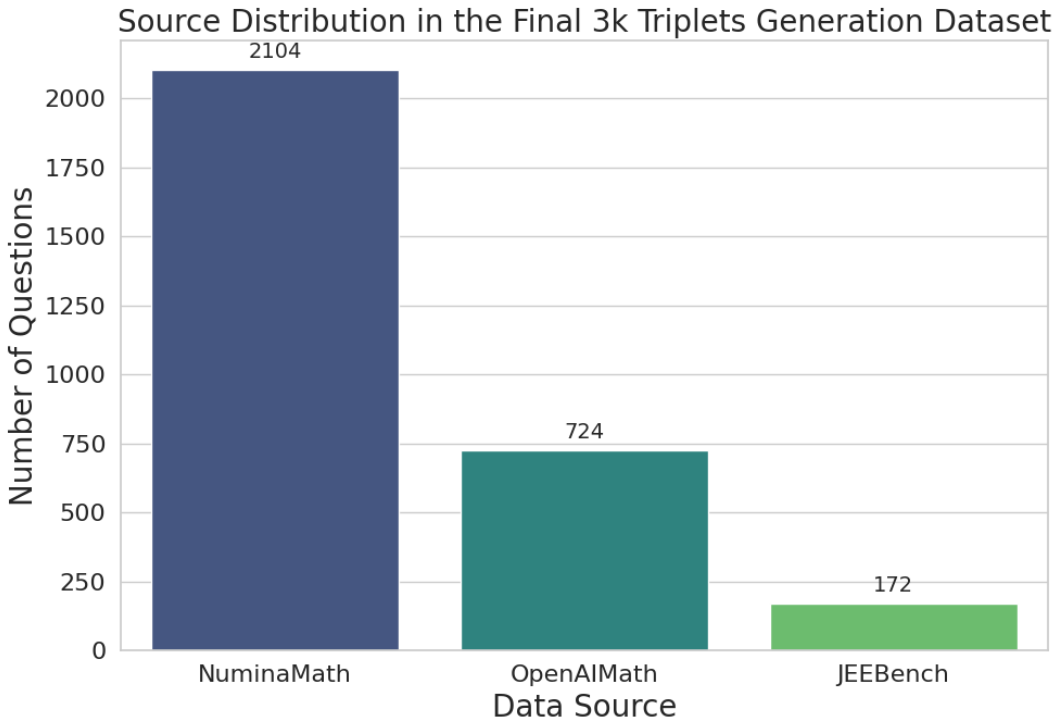

Figure 3.3: Source distribution in the final 3k dataset.

Box 1 : Prompt Template for Triplets Generation

You are a brilliant and meticulous math expert. Your task is to solve the given problem by providing a clear, step-by-step reasoning process. The reasoning should be concise and only include the necessary logical steps to reach the solution. Do not include unnecessary prose or overly detailed explanations of basic concepts.

After you have finished all reasoning steps, you must end your response with the final answer on a new line, in the EXACT format:
`#### <final_answer>`

Here is the problem:

—

{*question*}

The design of this prompt ensures that the teacher model can automatically parse and split the reasoning and generated_answer with a clean separator.
After 3,000 triplets were obtained from each teacher model, we designed an automated evaluation system based on a different commercial model to select data with the highest quality for subsequent distillation. The two sets of triplets generated by the teacher models were scored and screened individually. In the specific implementation, we selected Google Gemini 2.5 Pro model [Goo25] as an independent third-party 'judge' for the evaluation process. The judge model was also invoked via API. Input data, including the question, ground-truth answer, reasoning trace and actual answer

generated by the teacher model, and a structured scoring prompt were provided to the judge. It scored each sample based on the detailed scoring criteria from two dimensions: answer correctness and reasoning quality. Then it returned the result for each sample in JSON format. The detailed prompt is in Box 2.

Box 2 : Prompt Template for Triplets Quality Scoring

Role: You are an AI assistant specializing in evaluating mathematical reasoning. Your response MUST be ONLY a valid JSON object.

Task: Evaluate a math problem solution with two scores:

1. **Correctness:** 0 (wrong answer) or 1 (correct answer)
2. **Reasoning quality:** 1-10 scale based on logical soundness and clarity

Input: {`"input_json"`}

Evaluation Rules:

1. **Correctness (`is_correct`):**
   - Compare `generated_answer` with `final_answer`.
   - Score **1** if semantically equivalent (e.g., "15" = "15.0", "C" = full option C text).
   - Score **0** if factually different.
2. **Reasoning Quality (`reasoning_quality`):**
   Rate 1-10 based on:
   - Logical flow and correct steps (primary factor).
   - Clear explanation and organization.
   - Computational accuracy in reasoning process.

   Detailed Quality scale:
   - 8-10: Excellent - Clear, complete, logically sound.
   - 6-7: Good - Mostly clear with minor issues.
   - 4-5: Adequate - Has flaws but generally reasonable.
   - 2-3: Poor - Major logical errors or very unclear.
   - 1: Very poor - Severely flawed or incomprehensible.

Important: Even if final answer is wrong, reasoning can still score high if the process is sound.

Output Format (exactly this structure, no other text):
`{"is_correct": 0, "reasoning_quality": 5}`

After the judge model scored both 3,000-triplet datasets generated by the two teacher models, we selected only the 'perfect samples' with an answer correctness score of 1 and a reasoning quality score of 10 into the final candidate pool. Statistical results showed that GPT-4.1 generated 1,836 perfect samples, while Claude Sonnet 4 generated 1,991. Following the strategy in the previous paper [MYS+25], we also decided to use a very small-scale but good-quality training dataset with only a sample size of 1,000 for distillation in this study. Therefore, we performed stratified sampling based on data source again to reduce the total sample size to 1,000.

3.3 Knowledge distillation

After constructing the two Q-R-A triplet training datasets, this study carried out the first research objective: to transfer the reasoning capabilities from teacher models to an open-source student model through knowledge distillation, and observe the performance. As stated in the first chapter, we selected the Qwen2.5-32B-Instruct as the 'student model' for this experimental stage. This model has been chosen as the baseline model in many distillation related studies [LYX+25][BHA+24].
Fine-tuning a 32B parameter model is computationally intensive. To manage these resource demands, our training process utilized eight H100 GPUs (80GB VRAM each) [NVI25] and also employed QLoRA (Quantized Low-Rank Adaptation), a parameter-efficient fine-tuning (PEFT) technique. QLoRA [DPHZ23] integrates model quantization with Low-Rank Adaptation (LoRA) [HSW+22], which significantly reduces hardware requirements. This method enables the fine-tuning of large-scale models by updating only a small subset of additional parameters rather than the entire model.
To effectively transfer the reasoning patterns from the teacher models to the student, each of the 1,000 samples in the distillation dataset was structured into a standardized format. We employed a consistent prompt template that framed each sample as a prompt-question-answer triplet, which clearly defined the task and the expected output structure for the student model to learn. The detailed template is shown in Box 3.

Box 3 : Prompt Template for Training Data Formatting of Distillation

Instruction:
Solve the following math problem with step-by-step reasoning.

Problem:
{*question*}

Solution:
{*reasoning*}

Answer:
{*final_answer*}

The distillation process was conducted using the open-source SFTTrainer [vW+22b] from the trl library. While training, 1,000-size data generated by each teacher model was divided into training and evaluation datasets at a 9:1 ratio.The training metrics (loss and accuracy) were computed on

the evaluation dataset of 100 samples at 20 checkpoints throughout the 5 epochs. The checkpoint with the highest evaluation accuracy was saved. The key training hyperparameter are detailed in Table 3.2.

Table 3.2: Key Hyperparameters for Distillation

Hyperparameter	**Value**
Model and Fine-tuning Method	
Student Model	Qwen2.5-32B-Instruct
Fine-tuning Method	QLoRA (4-bit NF4 Quantization)
Max Sequence Length	8192
Training Parameters	
Epochs	5
Per-Device Batch Size	2
Gradient Accumulation Steps	8
Effective Batch Size	**16**
Learning Rate	1e-5
LR Scheduler	Cosine
Optimizer	Paged AdamW (8-bit)
Precision	bfloat16
Warmup Ratio	0.05
Weight Decay	0.01
LoRA Parameters	
LoRA Rank (r)	32
LoRA Alpha (α)	64
LoRA Target Modules	all-linear
LoRA Dropout	0.05

3.4 Post-Distillation Fine-Tuning with Reinforcement Learning (GRPO)

As pointed out by DeepSeek article [Gea25], subsequent RL training on models fine-tuned by distillation has the potential for further exploration. Therefore, in the final stage of this study, we selected the GRPO to further train the QLoRA fine-tuned models obtained in the distillation SFT stage. GRPO [PJ25] is an advanced policy gradient method designed to improve sample efficiency. Unlike traditional PPO algorithms, GRPO no longer relies on complex external reward models to assign absolute scores to individual outputs. Instead, it allows the model to generate a group of multiple candidate responses for each input. The reward signal is then derived from the relative ranking of these candidates, indicating which output within the group are better than the others. This approach can improve training stability and reduce computational costs, as it eliminates the need to train an independent large-scale reward model.

In practice, we built a hybrid reward model again around the Gemini 2.5 Pro API. Its scoring criteria is as follows:

- **Output Format Compliance Assessment:** A local evaluator performs a quick check on the text generated by the model to determine whether it adheres to the output generation formats (with <think>...</think> and <answer>...</answer> tag) predefined in the prompt in Box 4. If the output includes both the <think> and <answer> tags, it receives 1.0 points; if it includes only one of them, it receives 0.5 points; if neither is present, it receives 0.0 points.

Box 4 : Prompt Template for GRPO Generation

You are an AI assistant specializing in math. Solve the following problem step-by-step.

IMPORTANT FORMAT REQUIREMENTS:

1. Show your reasoning steps inside `<think>...</think>` tags (be concise, brief, essential steps only).
2. Enclose your final answer inside `<answer>...</answer>` tags.

Problem:
{*question*}

- **Output Answer Accuracy Assessment:** This assessment is also conducted by a local evaluator, primarily based on the equivalence of symbolic mathematical expressions and the exact matching of numerical and string values.
- **Gemini as the final judge:** When the local evaluator cannot determine the correctness of the answer (for example, due to complex answer formats or unit conversions), Gemini 2.5 Pro will intervene to perform a semantic-level comparison between the answers and provide a final correctness judgment. This criteria was passed to Gemini 2.5 Pro through the prompt shown in Box 5.

Box 5 : Prompt Template for Gemini-based Reward Model

Role: You are a professional math grader.

Task: Evaluate if the student's answer is mathematically equivalent to the reference answer.

Evaluation Rules:

- Focus ONLY on the content inside `<answer>...</answer>` tags.
- Score **1** if mathematically/semantically equivalent.
- Score **0** if different or incorrect.
- For multiple choice: exact letter match required.

Problem: {*question*}
Reference answer: {*ref*}
Student answer (from `<answer>` tags): {*pred_ans*}

Respond ONLY with: 1 or 0

The final reward was calculated using a weighted formula:
$Reward = 0.8 * AnswerCorrectnessScore + 0.2 * FormatComplianceScore$.
This formula is designed to guide the model to prioritize answer accuracy while encouraging it to 'think' by providing a clear reasoning format.
We originally planned to use 3,000 samples for GRPO training on 8 H100 GPUs. However, after practical testing, we found out that the GRPOTrainer [vW+22a] based on the TRL library has some historical issues in distributed training that are difficult to resolve, and the training time and cost are both quite high. Therefore, we decided to only construct a very small GRPO training dataset from the raw data (questions used in distillation training excluded) containing 100 samples. Training was conducted on a single H100 GPU with 80GB of VRAM.
For hyperparameters setup, we adopted a relatively low learning rate (LR) of 5e-6 to ensure the stability. We also introduced a KL Penalty Beta of 0.1 to prevent the model from deviating too far from the baseline reasoning capabilities it learned during the distillation phase. We set the core GRPO parameter Group Size to 3, which means for each question, three candidate answers will be generated . Additionally, we adopted a Temperature of 0.8 and a top-p (p=0.95) sampling strategy to encourage the model to explore diverse answers. The entire 100 samples were used to train in one complete epoch, with the parameters of the model updated every three steps. A detailed list of all hyperparameters is provided in Table 3.3.
To record the training effect, we tracked two key metrics at each step of the training process:

- **Training Loss:** used to observe the convergence trend of optimization;
- **Mean Reward:** the average reward of the three answers generated for the question of the current step, used to assess whether the model is improving.

Table 3.3: Key Hyperparameters for GRPO Reinforcement Learning

Hyperparameter	**Value**
Base Model and Method	
Starting Policy	SFT-tuned Qwen2.5-32B (QLoRA)
RL Algorithm	GRPO (Group-wise Reward Policy Optimization)
GRPO / Reinforcement Learning Parameters	
Total Train Epochs	1
Learning Rate	5e-6
Per-Device Batch Size	1
Gradient Accumulation Steps	3
Effective Batch Size	**3**
KL Penalty Beta (β)	0.1
Optimizer	Paged AdamW (8-bit)
Precision	bfloat16
Generation Parameters (for rollouts)	
Group Size (k)	3
Max New Tokens	1500
Max Prompt Length	512
Decoding Strategy	Top-p Sampling (p=0.95)
Temperature	0.8

3.5 Performance evaluation on benchmarks

After all the training was completed, we evaluated the performance of seven models, including: two teacher models (GPT-4.1 and Claude Sonnet 4); the baseline student model (Qwen2.5-32B-Instruct); two models fine-tuned via distillation (Qwen2.5-32B-SFT-GPT and Qwen2.5-32B-SFT-Claude); and two models further trained with GRPO (Qwen2.5-32B-SFT-RL-GPT and Qwen2.5-32B-SFT-RL-Claude).
Specifically, we selected two widely used open-source mathematical benchmarks to evaluate model performance:

- **GSM8K:** to evaluate the models' basic multi-step reasoning ability; For cost control, we randomly selected 200 questions from a total of 1,319 test questions.
- **AIME 2024:** challenging competition questions to test the models' capacity to solve complex problems. This dataset contains a total of 30 questions.

During the evaluation, all models were given the same unified prompt (see details in Box 6). To balance the creativity of reasoning with the stability of results, the temperature was set to 0.5. Since this strategy introduced a certain degree of randomness, we allowed the models to generate two independent answers for each question and used the average accuracy to calculate the final score to assess the models' actual capabilities. In practice, the teacher models were evaluated via

API calls, while the student models completed all the tests on servers deployed with 1–3 H100 GPUs.

Box 6 : Unified Prompt for Benchmark Evaluation

You are an AI assistant specializing in math.

Problem:
{*question*}

Please solve this step-by-step. Show your reasoning and calculations (be concise, brief, essential steps only). Put your final numerical answer after `####`

Chapter 4

Results

In this Chapter, we will present the results of the measurements of the experiments.

4.1 Distillation Metrics

This section presents the key metrics of the student models during distillation with GPT-4.1 and Claude Sonnet 4 as teacher models. Figure 4.1 shows the changes in Training Loss, while Figure 4.2 illustrates the changes in the models' average Mean Token Accuracy on the evaluation dataset. It is important to note that Mean Token Accuracy is calculated token by token, which means that every single token predicted by the student model is compared with the corresponding token in the evaluation set's ground truth. This metric measures the model's fidelity in accurately reproducing the teacher's output sequence, including both the reasoning trace and the answer components.
Figure 4.1 illustrates the evaluation loss during distillation. Both models show a clear downward trend across the five epochs, indicating effective learning. Notably, the student model trained on GPT-4.1 generated data consistently achieved a lower loss than that trained on data generated by Claude Sonnet 4. By the final epoch, the GPT-4.1 trained model converged to a loss of approximately 0.39, while the Claude sonnet 4 trained model concluded at around 0.42.
As shown in Figure 4.2, both models show a continuous upward trend across the evaluation dataset during the training progresses, indicating the improvement in the models' generalization capabilities. Mirroring the changes in loss, the model trained on GPT-4.1 data consistently outperformed the model trained on Claude Sonnet 4 data in accuracy across all epochs as well. By the end of the distillation, the GPT-distilled model achieved a peak accuracy exceeding 89%, with Claude-distilled surpassing 88%. Notably, the GPT-distilled model already reaches its peak accuracy around the third epoch, while the Claude-distilled model shows a more gradual and sustained upward trend.

4.2 Reinforcement learning (GRPO) Metrics

This section presents the detailed metrics of two student models trained by different teacher models during the subsequent RL (GRPO) phase. Figures 4.3 and 4.4 plot the Moving Average Reward and Moving Average Loss over 100 training steps (each for one mathematical problem), with a window size of 10 steps.

Figure 4.1: Comparison of training loss during the distillation phase. The student model trained on data generated by GPT-4.1 consistently achieved a lower loss than the model trained on Claude Sonnet 4 data.

As illustrated by the moving average reward in Figure 4.3, the learning process can be divided into three distinct phases. Within the initial 20 steps, the average rewards of both models show an overall upward trend, reaching a peak of 0.8 to 0.9. Subsequently, from approximately step 20 to 80, both models experience a decline phase that starts gradually and then accelerates into a steep drop, reaching their lowest reward points (approximately 0.3) near steps 85. In the final training phase (steps 85–100), a strong recovery is observed as both models' average rewards trend upward again. Throughout the GRPO training process, the reward curves of the two models were remarkably similar. However, a notable divergence occurs in the final stage, where the model distilled from GPT-4.1 demonstrates a stronger recovery, ultimately ending up with an average reward of approximately 0.85, higher than the approximately 0.75 achieved by the model distilled from Claude Sonnet 4.
As shown in Figure 4.4, the moving average loss during GRPO training presents a contrast to the distillation loss curve. Instead of expected monotonic decline, the loss curves for both models fluctuate continuously. A particularly noteworthy phenomena can be observed between steps 60 and 80, where both models show a counterintuitive upward trend in loss, peaking near step 80. After this peak, a dramatic decline occurred immediately. Overall, the loss curves of the two models are highly correlated, with their trends overlapping for most of the time, and neither showing a clear advantage.

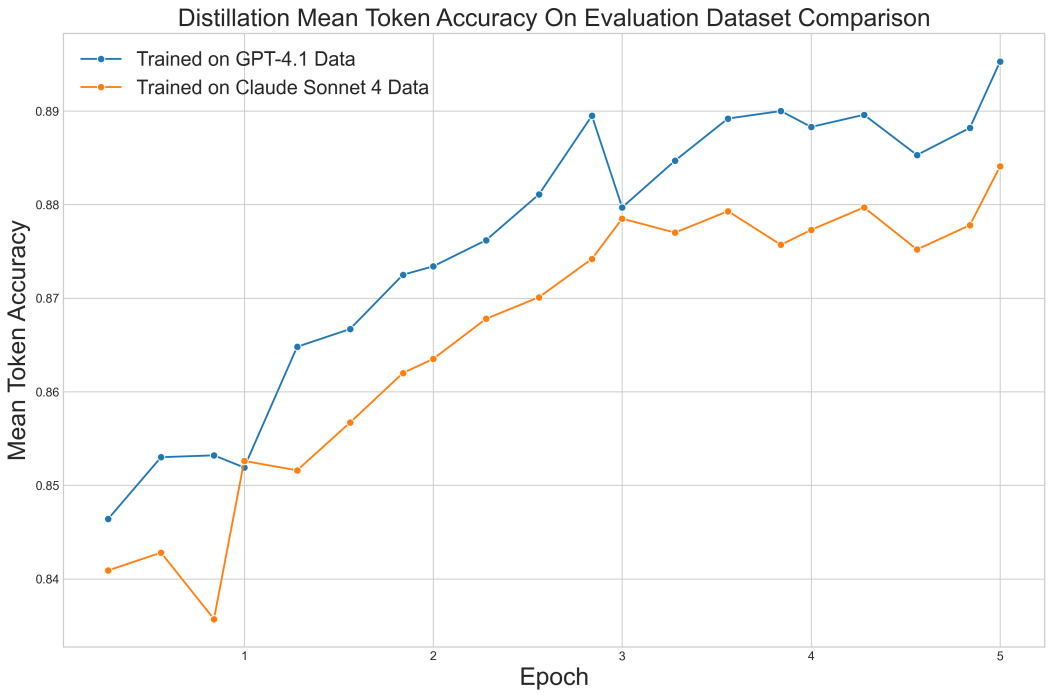

Figure 4.2: Comparison of mean token accuracy on the evaluation dataset during the distillation phase. The model distilled from GPT-4.1 data reached a higher peak accuracy (approx. 89.5%) compared to the model trained on Claude data (approx. 88.4%), suggesting superior data quality for this task.

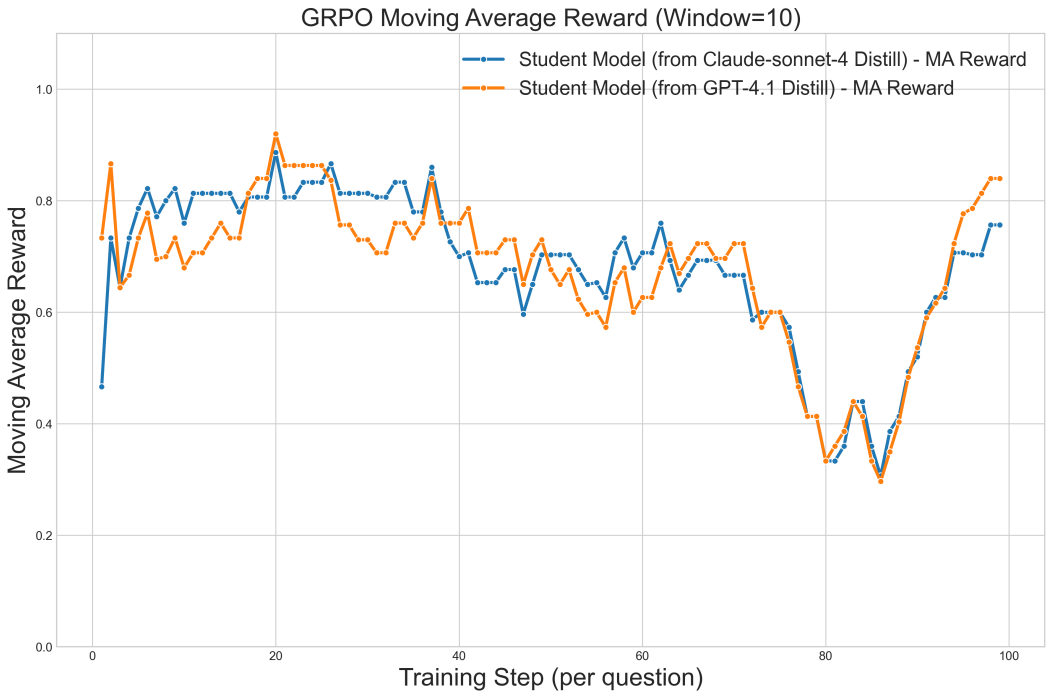

Figure 4.3: The reward curves of GRPO training show three distinct phases: an initial upward trend, a period of decline, and a final recovery, with the GPT-4.1-distilled model achieving a higher final reward.

4.3 Benchmark scores

The final evaluation results are presented in Table 4.1, which compares the performance of seven models across the GSM8K and AIME 2024 benchmarks. The teacher models, GPT-4.1 and Claude Sonnet 4, established the best performance, while the baseline student model Qwen2.5-32B-Instruct scored 47.8 on GSM8K and 1.7 on AIME.

After the initial SFT distillation, both student models showed improvement on GSM8K, with the GPT-4.1-distilled model having a slight higher score at 49.5. However, the subsequent GRPO training produced a performance reversal. The Claude sonnet 4-distilled model's score was further enhanced to 49.8, making it the top-performing student model. In contrast, the GPT-4.1-distilled model's performance regressed to 46.3, even falling below the initial baseline.

Meanwhile, on the more challenging AIME 2024 benchmark, neither the distillation nor the GRPO stage yielded any measurable performance gains for the student models.

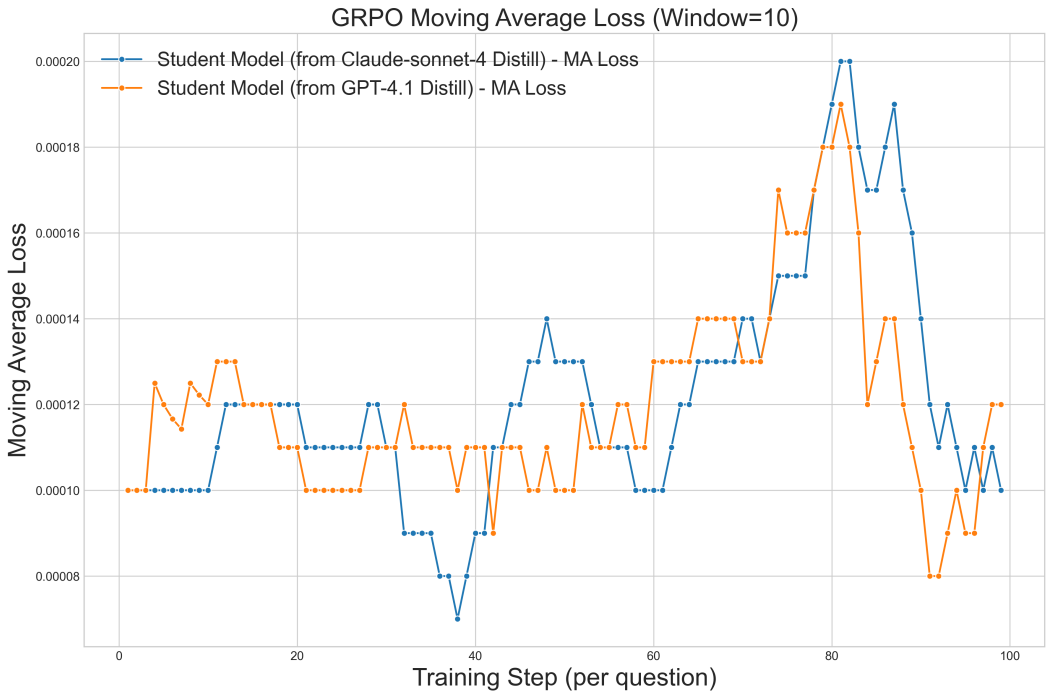

Figure 4.4: Moving average loss during the GRPO fine-tuning phase. Unlike the loss of distillation stage, the loss curves show significant fluctuations rather than a monotonic decline, with neither model showing a consistent advantage.

Table 4.1: Comparison of Model Performance on Mathematical Reasoning Benchmarks. All scores are reported as mean accuracy in percentage (%). The best score in each column is highlighted in bold.

Model	**GSM8K**	**AIME 2024**
Teacher Models (Performance Benchmark)		
GPT-4.1	93.5	**36.7**
Claude Sonnet 4	**94.5**	35.0
Student Model (Qwen2.5-32B-Instruct)		
Baseline (Untrained)	47.8	1.7
After Distillation (SFT)		
SFT (distilled from GPT-4.1)	49.5	1.7
SFT (distilled from Claude Sonnet 4)	49.3	1.7
After SFT + GRPO Alignment		
SFT (from GPT-4.1) + GRPO	46.3	1.7
SFT (from Claude Sonnet) 4 + GRPO	**49.8**	1.7

Chapter 5

Discussion

In this chapter, we will discuss the possible reasons behind the results shown in the previous chapter.

5.1 Knowledge Distillation Effect

The experimental results of the knowledge distillation phase proved its potential as an effective knowledge transfer approach. Trained with only 1,000 good-quality Q-R-A triplets, the student models achieved a stable improvement of 1.5%–1.7% in performance on the GSM8K benchmark compared to the baseline model. This indicates that even with limited data, the student models can still successfully learn reasoning ability from the teacher model.

However, the study results also demonstrate that the improvement of distillation can be limited, especially when small models are facing complex tasks. First, our distilled student model did not outperform the teacher model. On GSM8K, the student model achieved a maximum accuracy of approximately 50%, which is significantly lower than the teacher models' 93-94% accuracy. On the more challenging AIME 2024 benchmark, this gap was further amplified with the student models' performance being close to zero. This phenomenon aligns with the theory proposed by Li et al. [LYX+25]: when the teacher model is too powerful, its complex, long-chain reasoning processes may be difficult for small models with limited capacity to fully digest. This suggests that simple knowledge distillation can be effective, but still insufficient for small models to obtain the similar capability of state-of-the-art large models.

Additionally, the limited improvement (approximately 2%) achieved with 1,000-question dataset in this study contrasts sharply with the findings of Muennighoff et al. [MYS+25], who achieved a more significant performance leap with the same data scale. We speculate that this difference could stem from the following core factors: first, their dataset was not simply a collection of "perfect examples," but specifically selected challenging problems that the student model had previously answered incorrectly; second, they employed a strategy called "Budget Forcing", which forced the model to extend its thinking time to conduct self-correction.

Ultimately, these results indicate that the most efficient approach to improve small model's reasoning capability by distillation is not simply to let it emulate the teacher model, but rather to implement a targeted strategy that directly corrects the student's deficiencies.

5.2 Selection of teacher models

When comparing the effectiveness of GPT-4.1 and Claude Sonnet 4 as teacher models, we observed this phenomenon: a powerful teacher does not necessarily result in a good students.
It can be seen from the distillation metrics that GPT-4.1-distilled student model achieves a higher peak accuracy on the evaluation dataset (89.5% vs. 88.4%). However, in the final benchmark test, the performance of the student models trained by both teacher models is nearly identical (49.5% vs. 49.3%) on GSM8K. In the subsequent GRPO fine-tuning phase: the GSM8K score of Claude-sonnet 4-distilled model improved to 49.8%, becoming the best-performing student model; however, the GPT-4.1-distilled model showed a performance decline instead of improvement by dropping to 46.3%.
This phenomenon could be explained as follows. GPT-4.1's reasoning style may be more complex, abstract, and diverse. Hence, the student might overfit to this during the distillation phase. However, the GRPO training requires active exploration and optimization. In this case, the complex distribution of training data generated GPT-4.1 can become a distraction, causing the model to deviate from the correct problem-solving path while optimizing reward signals, which results in a loss of capability.
On the other hand, Claude Sonnet 4's reasoning trace may be closer to the student model's own capability boundaries and internal representation distribution. This enables the student model not only to learn more effectively during the distillation, but also to optimize further in subsequent reinforcement learning based on the capability it already developed.
This finding might corroborate the view of Bansal et al. [BHA+24]: overly powerful teacher models sometimes reduce learning efficiency due to significant differences between their output distributions and those of the student model. This also aligns with the research of Koo et al. [KHK+24], who similarly emphasize that the distribution matching between teacher and student models is critical for the stability and final effectiveness of distillation.

5.3 Reinforcement Learning (GRPO)

While exploring whether RL (GRPO) can further improve distilled models' performance, we observed this phenomenon: under our current resource-constrained experimental setup, GRPO only provides very limited and unstable performance gains, and in some cases, even negative effects. We have proposed the following hypotheses:

- **Training Data Scarcity:** Attempting to fine-tune a policy with only 100 training samples likely led to severe overfitting. The model may have simply memorized responses rather than learning generalizable reasoning strategies.
- **Limited Candidate Pool:** With a group size of only three candidate answers per question, the opportunity for the reward model to identify a real high-quality one was statistically limited, thereby making it harder for the model to improve.

Comparing our study with the successful large-scale experiments by the DeepSeek team [Gea25], we infer that the efficacy of post-distillation RL fine-tuning is highly rely on substantial training resources, including large datasets and meticulous hyperparameter optimization. When the resources

are limited, applying GRPO may introduce risks of overfitting and performance degradation that outweigh the potential benefits.

Chapter 6

Conclusion

In this chapter, we will answer the problem statement and research questions brought up at the beginning of this study.

6.1 Outperforming Large Models

RQ1: Can small models trained via distillation with limited data outperform large models in mathematical reasoning?
Under our current constrained setup, the answer of this research question is negative. Although the 32B-parameter student model did achieve a performance improvement through distillation (accuracy score on GSM8K increased by approximately 1.7% from the baseline), this was still insufficient to fill in the capability gap. A significant performance disparity of about 44% can be observed when compared to the teacher models. This capability gap was even more obvious on the more challenging benchmark AIME 2024, where the student models showed no improvement over the baseline at all. Their performance remained at 1.7%, far below the 35-37% scores achieved by the teacher models. Hence, our study results demonstrate that knowledge distillation does not necessarily enable a small model to achieve a level of mathematical reasoning capability comparable to that of commercial LLMs. The effect of distillation highly depends on the actual conditions.

6.2 Optimal Teacher Model

RQ2: Is GPT or Claude a better teacher model for distillation?
Regarding this research question, our conclusion is that the optimal teacher model is not necessarily the most powerful one, but the one whose outputs align better with the student's capacity and distribution. This was evidenced by the model performance after the RL stage in this study. While the GPT-4.1-distilled model showed better metrics during distillation, the model distilled from Claude Sonnet 4 proved to be more robust and ultimately achieved a higher score on benchmark after GRPO fine-tuning. We attribute this to Claude's output style and distribution being closer to that of the student model, which provided a more effective learning foundation for it.

6.3 Impact of Reinforcement Learning

RQ3: Can subsequently fine-tuning a distilled student model with RL further improve the performance?
Regarding this question, the findings of this study provide a negative answer under our experimental conditions. The RL fine-tuning conducted with a training set of only 100 samples failed to produce significant and consistent performance improvements. The training process itself proved to be highly unstable with large fluctuations throughout the steps. Furthermore, the impact of GRPO was highly dependent on the model's initial state after distillation. While the model distilled from Claude Sonnet 4 remained the similar level of performance after GRPO, the model distilled from GPT-4.1 showed a notable performance degradation, even falling below the baseline. This demonstrates that, with a small-scale training setup, GRPO is not a reliable method for enhancement and may even cause a risk of degrading the model's previously learned capabilities.

6.4 Revisiting the Problem Statement

PS: Can reinforcement learning improve distillation for trainning high-performance small LLMs?
Finally, we will address the main problem statement of this study. Based on the evidence gathered above, our conclusion is that under significant data and resource constraints, introducing a subsequent RL fine-tuning stage does not reliably improve the performance of distilled models and may even pose a risk of degradation. The possible reasons for this outcome are as follows. Firstly, the RL training on an extremely small dataset proved highly unstable and can led to overfitting, which will harm the model's generalization capabilities. Secondly, a limited candidate pool in GRPO setup can restrict the reward model's ability to find optimal responses, thus making it more difficult for the model to improve itself.

6.5 Limitations and Further Research

The conclusions of this study are subject to several key limitations. Firstly, the scale of the training data was highly constrained. Both the 1,000-sample distillation training dataset and the 100-sample GRPO training dataset are of a limited size, which likely restricted the model's ultimate generalization capabilities. Secondly, the GRPO phase was also constrained by limited computational resources being only conducted on a single H100 GPU. This restricted the number of training iterations and the breadth of the hyperparameter search, which means the optimal configuration for the GRPO algorithm may not have been found in this study. Furthermore, the strategy of evaluating model performance for this study mainly relies on the final answer's accuracy rather than then quality of the reasoning trace, such as its logical coherence or step-by-step correctness. Finally, the small size of both evaluation benchmarks means that the observed performance differences between models may not be statistically significant, and thus should be interpreted with caution.
Based on the findings of this study and the identified limitations above, we believe that the following research directions are worth further exploration in the future. First, the scale and diversity of training data should be expanded. This is not only about building larger datasets covering more problem types and difficulty levels, but also exploring multi-teacher ensemble distillation strategies.

Additionally, it is also crucial to establish more targeted training datasets based on the token distribution characteristics and weaknesses of the student model itself.
Second, in terms of evaluation strategies, we suggest future work to develop a more comprehensive evaluation model. This model should not only be limited to evaluate the accuracy of the final answer, but also introduce a fine-grained analysis of the reasoning quality, such as the rigor of logic, the accuracy of reasoning steps, and the interpretability, to better measure the model's ability. At the same time, benchmarks with larger scale should be employed to better ensure the statistical significance.
Finally, our budget issue severely limited the available computational resources for this study. Therefore, future experiments should allocate more powerful computational resources if possible. This will support more extensive and systematic hyperparameter optimization of RL algorithms like GRPO, enabling a more accurate assessment of their true potential and optimal configurations in post-distillation scenarios.

Bibliography

[AI-24] AI-MO. NuminaMath-CoT Dataset. https://huggingface.co/datasets/AI-MO/NuminaMath-CoT, 2024. Accessed on August 16, 2025.

[Ant23] Anthropic. Claude 2. https://www.anthropic.com/news/claude-2, July 2023. Accessed on August 16, 2025.

[Ant25] Anthropic. Claude sonnet language model. https://www.anthropic.com/claude/sonnet, 2025. Accessed on August 15, 2025.

[BGMMS21] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623. Association for Computing Machinery, 2021.

[BHA+24] H. Bansal, A. Hosseini, R. Agarwal, V. Q. Tran, and M. Kazemi. Smaller, weaker, yet better: Training llm reasoners via compute-optimal sampling. *arXiv preprint arXiv:2408.16737*, 2024.

[BPW+12] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[CJLW22] Wenhu Chen, Boshen Jiang, Yan-Rong Li, and William Wang. Program of thoughts prompting: Disentangling computation from reasoning for more accurate math problem solving. *arXiv preprint arXiv:2211.12588*, 2022.

[DAI24] DAIR-IITD. JEEBench Dataset. https://github.com/dair-iitd/jeebench, 2024. Accessed on August 16, 2025.

[DPHZ23] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36, 2023.

[Gea25] Daya Guo and et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[Goo25] Google Cloud. Gemini 2.5 pro model documentation. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro, 2025. Accessed on August 18, 2025.

[GT24] Google Gemini Team. Gemini 1.5: Unlocking multimodal understanding across long contexts. Technical report, Google, February 2024. https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf.

[HBK+21] 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.

[HSW+22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[Hug21] Hugging Face. GSM8K Dataset. https://huggingface.co/datasets/openai/gsm8k, 2021. Accessed on August 16, 2025.

[Hug24a] Hugging Face. Transformers Documentation - Gemma 2. https://huggingface.co/docs/transformers/en/model_doc/gemma2, 2024. Accessed on August 16, 2025.

[Hug24b] Hugging Face H4 Team. AIME 2024 Dataset. https://huggingface.co/datasets/HuggingFaceH4/aime_2024, 2024. Accessed on August 16, 2025.

[IBM23] IBM. What are large language models (llms)? *IBM*, 2023. Retrieved August 23, 2023, from https://www.ibm.com/think/topics/large-language-models.

[KHK+24] J. Koo, Y. Hwang, Y. Kim, T. Kang, H. Bae, and K. Jung. SWITCH: Studying with teacher for knowledge distillation of large language models. *arXiv preprint arXiv:2410.19503*, 2024.

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[KWBH24] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2408.00127*, 2024.

[LMS23] LMSYS Org. Vicuna-13B-v1.3 Model Card. https://huggingface.co/lmsys/vicuna-13b-v1.3, 2023. Accessed on August 16, 2025.

[LYX+25] Y. Li, X. Yue, Z. Xu, F. Jiang, L. Niu, B. Y. Lin, and R. Poovendran. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025.

[MYS+25] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Fei-Fei Li, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[NKQ+23] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[NVI25] NVIDIA. Nvidia h100 tensor core gpu. https://www.nvidia.com/en-us/data-center/h100/, 2025. Accessed on August 18, 2025.

[Ope21] OpenAI. Grade School Math Dataset (GSM8K). https://github.com/openai/grade-school-math, 2021. Accessed on August 16, 2025.

[Ope23] OpenAI. Gpt-4 technical report. Technical Report 2303.08774, arXiv, 2023. https://arxiv.org/abs/2303.08774.

[Ope24a] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, May 2024. Accessed on August 16, 2025.

[Ope24b] OpenRouter. Openrouter - the ai router for llms. https://openrouter.ai/, 2024. Accessed on August 18, 2025.

[Ope25] OpenAI. Gpt-4.1 model documentation. https://platform.openai.com/docs/models/gpt-4.1, 2025. Accessed on August 15, 2025.

[OWJ+22] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[PJ25] Lei Pang and Ruinan Jin. On the Theory and Practice of GRPO: A Trajectory-Corrected Approach with Fast Convergence, 2025.

[Qwe25] Qwen Team. Qwen2.5-32B-Instruct Model Card. https://huggingface.co/Qwen/Qwen2.5-32B-Instruct, 2025. Accessed on August 15, 2025.

[RGG+23] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Tworkowski, Marie-Anne Lachaux, Thibaut Lavril, Izabela Pytlarz, Mieszko Szafraniec, Guillaume Lample, Eleonora Hambro, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[RSM+23] 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. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[SSS23] K. Shridhar, A. Stolfo, and M. Sachan. Distilling reasoning capabilities into smaller language models. pages 7059–7073, 2023.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[TBL+23] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, and Clément Fourrier. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

[vW+22a] Leandro von Werra et al. GRPOTrainer: Group-wise Reward Policy Optimization component of the TRL Library. https://huggingface.co/docs/trl/main/en/grpo_trainer, July 2022. Accessed on August 19, 2025.

[vW+22b] Leandro von Werra et al. SFTTrainer: Supervised Fine-tuning component of the TRL Library. https://huggingface.co/docs/trl/en/sft_trainer, July 2022. Accessed on August 19, 2025.

[WWS+22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc 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.

[YZW+25] H. Yin, Y. Zhao, M. Wu, X. Ni, B. Zeng, H. Wang, and K. Zhang. Towards widening the distillation bottleneck for reasoning models. *arXiv e-prints*, 2025.

[ZLL+24a] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang. Distilling mathematical reasoning capabilities into small language models. *Neural Networks*, 179:106594, 2024.

[ZLL+24b] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.

[ZWF+25] Y. Zhang, L. Wang, M. Fang, Y. Du, C. Huang, J. Wang, and Q. Zhang. Distill not only data but also rewards: Can smaller language models surpass larger ones? *arXiv preprint arXiv:2502.19557*, 2025.

[ZZL+23] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.