Phi-4-Mini-Reasoning: Exploring the Limits of Small Reasoning Language Models in Math

Haoran Xu‡ Baolin Peng‡ Hany Awadalla Dongdong Chen Yen-Chun Chen Mei Gao Young Jin Kim Yunsheng Li Liliang Ren Yelong Shen Shuohang Wang Weijian Xu Jianfeng Gao Weizhu Chen

Microsoft

Abstract

Chain-of-Thought (CoT) significantly enhances formal reasoning capabilities in Large Language Models (LLMs) by training them to explicitly generate intermediate reasoning steps. While LLMs readily benefit from such techniques, improving reasoning in Small Language Models (SLMs) remains challenging due to their limited model capacity. Recent work by Deepseek-R1 (Luo et al., 2025) demonstrates that distillation from LLM-generated synthetic data can substantially improve the reasoning ability of SLM. However, the detailed modeling recipe is not disclosed. In this work, we present a systematic training recipe for SLMs that consists of four steps: (1) large-scale mid-training on diverse distilled long-CoT data, (2) supervised fine-tuning on high-quality long-CoT data, (3) Rollout DPO leveraging a carefully curated preference dataset, and (4) Reinforcement Learning (RL) with Verifiable Reward. We apply our method on `Phi-4-Mini`, a compact 3.8B-parameter model. The resulting **`Phi-4-Mini-Reasoning`** model exceeds, on math reasoning tasks, much larger reasoning models, e.g., outperforming DeepSeek-R1-Distill-Qwen-7B by 3.2 points and DeepSeek-R1-Distill-Llama-8B by 7.7 points on Math-500. Our results validate that a carefully designed training recipe, with large-scale high-quality CoT data, is effective to unlock strong reasoning capabilities even in resource-constrained small models.

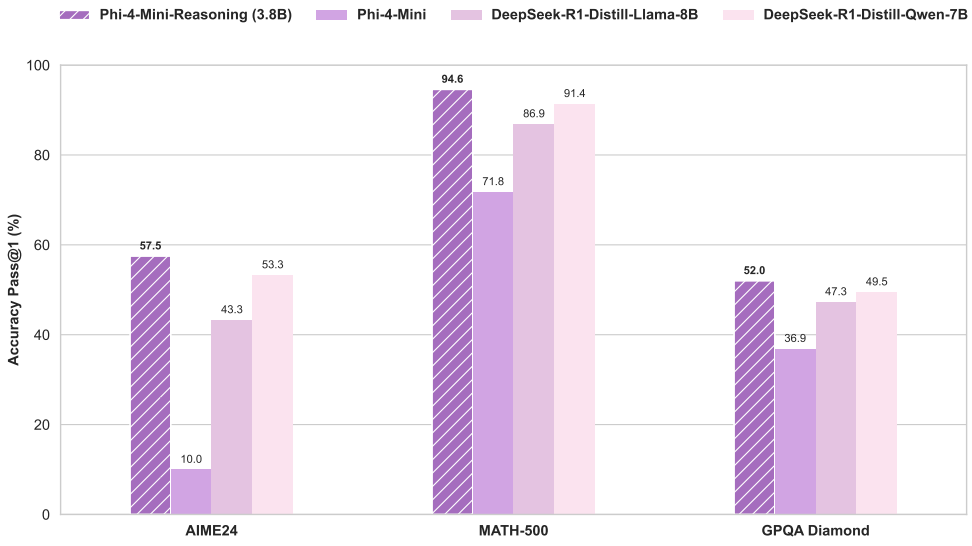


Figure 1: Math benchmark performance of Phi-4-Mini-Reasoning.

‡Equal Contribution. Except for the first and last two authors, the remaining authors are listed in alphabetical order.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across numerous natural language processing tasks, while their reasoning ability often deteriorates when confronting intricate, multi-step problems, where simply outputting an answer without intermediate steps leads to significant performance gaps (Wei et al., 2022). The Chain-of-Thought (CoT) approach addresses this challenge by explicitly prompting models to generate a sequence of logical steps prior to arriving at a final answer, thereby significantly enhancing their reasoning capacities (Kojima et al., 2022; Wei et al., 2022). Incorporating this reasoning process during inference has established the paradigm of test-time scaling, which further elevates performance in complex reasoning tasks (Snell et al., 2024; Welleck et al., 2024; OpenAI, 2024).

Enhancing reasoning abilities is inherently easier for larger LLMs due to their extensive capacity, whereas it remains challenging for Small Language Models (SLMs). Fortunately, Deepseek-R1 (Guo et al., 2025) indicates that non-logits-level distillation—effectively supervised fine-tuning (SFT) of SLMs using synthetic data generated by more capable models—can markedly improve SLM reasoning performance. For instance, such an approach could elevate MATH-500 (Lightman et al., 2023) accuracy of Llama-8B (Grattafiori et al., 2024) from 44.4% to 89.1% . Following this breakthrough, numerous efforts, including Bespoke-Stratos-7B (Labs, 2025) and OpenThinker-7B (OpenThoughts, 2025), have aimed to replicate these results. Despite this enthusiasm, debates persist regarding the primary focus of training. Deepscaler (Luo et al., 2025) suggests scaling RL like GRPO (Shao et al., 2024) for reasoning gains, while S1 and LIMO (Muennighoff et al., 2025; Ye et al., 2025b) emphasize the quality and diversity of reasoning datasets, revealing that even datasets as small as fewer than 1K examples can enhance reasoning performance.

Rather than focusing on isolated techniques that individually benefit training, we systematically explore a training paradigm specifically tailored for SLMs, where limited model capacity makes reasoning improvements particularly challenging. Our methodology consists of two stages of distillation, followed by rollout-based preference learning that also reuses wrong LLM-generated samples, and concludes with RL using a verifiable reward. Initially, we employ distillation as a mid-training mechanism to embed foundational reasoning capabilities. We then apply distillation again in a fine-tuning phase to further improve model generalization. During LLM rollout sampling for distillation, some incorrect outputs are typically discarded; however, we re-purpose these discarded samples to create a customized preference dataset, which is used for preference learning applied on top of the distilled model. Finally, we fine-tune the model using reinforcement learning with a verifiable reward signal based on final answer correctness. To ensure stable training, we introduce several targeted improvements, including prompt optimization, reward re-balancing via oversampling and filtering, and temperature annealing during exploration.

We validate our proposed approach using Phi-4-Mini (Microsoft et al., 2025), a compact 3.8-billion-parameter model, resulting in **`Phi-4-Mini-Reasoning`**, which outperforms other reasoning models nearly twice its size, such as DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B.

2 Background

Small language models (SLMs) have demonstrated significant potential for strong reasoning capabilities. For example, Qwen-1.5B can achieve 83.9% accuracy on the Math-500 (Lightman et al., 2023) benchmark simply by distilling 800K examples from DeepSeek-R1 (Guo et al., 2025). Distillation has emerged as a powerful tool to enhance the reasoning abilities of SLMs; however, the optimal distillation strategy for small models remains underexplored. Recent studies provide complementary insights: Luo et al. (2025) suggests that gradually increasing generation length via reinforcement learning can further improve

distilled models, while Muennighoff et al. (2025) and Ye et al. (2025b) emphasize that data diversity and quality, rather than quantity alone, are critical to success. Despite these advances, a comprehensive understanding of an effective distillation recipe for SLMs is still lacking. Moreover, naively applying isolated techniques can lead to degraded performance. For instance, directly distilling S1K (Muennighoff et al., 2025) or LIMO (Ye et al., 2025b) datasets onto Phi-4-Mini results in a significant drop in reasoning performance. This observation suggests that SLMs, due to their limited capacity, require substantially more carefully designed data and training strategies to develop robust reasoning capabilities compared to their larger counterparts. Detailed results illustrating this phenomenon are shown in Table 1.

Model	**AIME 2024**	**MATH-500**	**GPQA Diamond**
Phi-4-Mini	10.0	71.8	36.9
Phi-4-Mini + LIMO	6.7	57.8	24.8
Phi-4-Mini + S1K	3.0	47.0	26.3
Phi-4-Mini-Reasoning (with our full recipe)	**57.5**	**94.6**	**52.0**

Table 1: Pass@1 performance of Phi-4-Mini under different distillation settings. Naively using a small amount of high-quality data leads to significant performance degradation, highlighting the necessity of a comprehensive training recipe.

Hence, our goal is to develop a comprehensive and efficient recipe for training SLMs. We first observe that non-reasoning SLMs require an initial mid-training stage to absorb a large volume of reasoning trajectories before any additional techniques are applied. However, several key questions remain: How much mid-training data is necessary? What subsequent techniques–such as careful distillation, preference learning, or reinforcement learning–should be employed next? In this work, we systematically address these questions and propose a complete recipe for building high-performing reasoning SLMs.

3 Multi-Stage Continual Training for Reasoning

Here, we systematically introduce our complete training recipe rather than exploring individual components. Taking a pre-trained SLM as a base, we improve its formal reasoning capability by first performing a multi-stage continual training using a curated CoT reasoning dataset and then running RL with verifiable rewards.

3.1 Distillation as Mid-Training

In the first stage, we frame distillation as mid-training. Specifically, we train the base model with next token prediction on an extensive corpus of synthetic chain-of-thought (CoT) data, which covers questions from diverse domains and varying levels of difficulty. The CoT-style answers for these questions are sampled by the Deepseek-R1 model (Guo et al., 2025), after which we apply rejection sampling to retain only the correct answers. More details on our data generation methodology are presented in Section 4. We pair each question with its corresponding correct CoT answer and train the base model using the standard causal language modeling objective. We train the model under a *packing* mode, i.e., multiple short examples are packed in the same input sequence to increase training efficiency. The goal of this mid-training step is to equip the small base model with general CoT reasoning capabilities that are not explicitly learned during model mid-training. We find it effective to allow mid-training to iteratively use as much CoT training data as possible until model performance saturates on a validation dataset.

3.2 Distillation as Supervised Fine-tuning

After learning extensive and diverse reasoning chains, our next step involves selecting a compact, yet representative, subset from the mid-training dataset for subsequent fine-tuning. Fine-tuning is performed in a *non-packing* mode where we teach the model to decide where to stop generating. As it has been shown that higher-quality data can notably improve model performance and generalization capabilities and enable the model to better answer complex questions (Xu et al., 2024a; Zhou et al., 2023; Ye et al., 2025b; Muennighoff et al., 2025), we have constructed a combined dataset spanning diverse math domains, with difficulty levels exceeding the 'college level'. More details about data categorization are described in Section 4.

3.3 Rollout Preference Learning

In the previous two stages, the model is trained exclusively on accepted generations, filtering out rollouts containing incorrect answers. However, are the rejected rollouts entirely devoid of value? In this stage, we use rejected rollouts to enhance model performance. The quality of rejected data is important for preference learning, as pointed out by Xu et al. (2024b),. Specifically, incorrect responses with minor nuances compared to their correct counterparts provide effective candidates for constructing informative preference pairs. To ensure data quality, we retained the questions that are categorized as 'high-school' level math or above, determined by GPT-4o-mini (Achiam et al., 2023). The preference dataset is then constructed by designating correct answers as preferred rollouts and incorrect answers as dis-preferred rollouts for each question. Finally, we apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to the model:

$$J_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\Big], \tag{1}$$

where π_{ref} is the reference model, y_w and y_l are preferred and dis-preferred rollouts, respectively.

3.4 RL with Verifiable Reward

Although DPO improves the model's alignment and reasoning ability using curated preference pairs, DPO is limited as an offline learning method using a fixed dataset. To improve model's reasoning capability through online learning, we perform RL on the distilled and preference-trained model. In what follows, we describe the RL algorithms we have experimented and the RL training recipe.

Proximal Policy Optimization (PPO) PPO (Schulman et al., 2017) has been successfully applied to fine-tuning LLMs via RLHF. The algorithm employs a clipped surrogate objective to limit each policy update so that it stays close to the previous policy. This clipping mechanism avoids overly large importance sampling ratios, which both stabilizes learning and enhances sample efficiency. PPO seeks to maximize

$$J_{\text{PPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o_{\le t}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Bigg[\min\Big(r_t(\theta)\,\widehat{A}_t,\ \text{clip}\big(r_t(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\widehat{A}_t\Big)\Bigg], \tag{2}$$

where

$$r_t(\theta) = \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t})},$$

and q is sampled from the data distribution $\mathcal{D}$, ϵ controls the clipping range, and $\widehat{A}_t$ denotes the advantage estimate at time step t. To compute $\widehat{A}_t$, PPO uses the Generalized Advantage Estimator

(GAE) (Schulman et al., 2015). Given a value function V and a reward function R, the estimator is

$$\widehat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\,\lambda)^l\,\delta_{t+l}, \tag{3}$$

with the temporal-difference term

$$\delta_l = R_l + \gamma\,V(s_{l+1}) - V(s_l), \quad 0 \le \gamma, \lambda \le 1. \tag{4}$$

Group-based Relative Policy Optimization (GRPO) GRPO (Shao et al., 2024) estimates its baseline by comparing rewards within a batch of G model responses, reducing the critic's cost and improving model training stability. Concretely, for each question q, it samples a set of candidate responses G $\{o_i\}_{i=1}^{G}$ under the old policy $\pi_{\theta_{\mathrm{old}}}$, then computes their rewards $\{R_i\}_{i=1}^{G}$. The normalized advantage is computed as

$$A_i \;=\; \frac{R_i - \mathrm{mean}(R_1, \ldots, R_G)}{\mathrm{std}(R_1, \ldots, R_G)}, \tag{5}$$

GRPO then maximizes a clipped-surrogate objective, averaged over the group, with an additional KL-penalty toward a reference policy π_{ref}:

$$J_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\,\{o_i\}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\min\Big(r_i(\theta)\,A_i,\ \mathrm{clip}\big(r_i(\theta),\,1-\epsilon,\,1+\epsilon\big)\,A_i\Big)\;-\;\beta\,D_{KL}\big(\pi_\theta\|\pi_{\mathrm{ref}}\big)\right], \tag{6}$$

where ϵ is the clipping parameter and β weights the KL-penalty.

Verifiable Reward Reinforcement learning with verifiable reward (RLVR) has shown to be very effective in training models for various reasoning tasks (Guo et al., 2025; Ye et al., 2025a). Following prior work (Guo et al., 2025), the reward for a verifiable task is defined as a function of the accuracy of the model's final answer. Concretely,

$$R(\hat{y}, y) = \begin{cases} +1, & \text{if } \texttt{verify}(\hat{y}, y), \\ -1, & \text{otherwise.} \end{cases} \tag{7}$$

where y denotes the ground-truth answer and $\hat{y}$ the response of the model.

Our RL Recipe In our pilot study of applying GRPO to train our base model, we have observed three issues that affect the stability and effectiveness of model training.

1. **High Variance in Response Lengths** Although the base model, after mid-training, is already able to generate reasonable CoT responses, we have observed substantial variability in response lengths within the same GRPO sampling group. For the same prompt, positively rewarded responses ranged from approximately 12k to 20k tokens. Directly optimizing the model for the standard GRPO objective on such length-heterogeneous responses induces instability. Zhang et al. (2025) reports a similar phenomenon when training a model on both mathematical and coding tasks.

2. **Vanishing Gradients under Uniform Rewards** GRPO's reliance on advantage estimates makes it susceptible to the vanishing gradient problem of all sampled responses in a group receiving identical rewards, yielding zero variance in the returns. The DAPO framework (Yu et al.) addresses this problem by oversampling and filtering out prompts whose response accuracies are exactly 0 or 1, thereby preserving non-zero advantage signals. However, we find in our experiments that we need to address the following two problems when applying DAPO to our model:

(i) The model is sensitive to intra-group length discrepancies: responses with intermediate accuracies (e.g., 0.1 or 0.9) still provoke unstable gradient magnitudes due to response-length variance.

(ii) For difficult math tasks, attaining even a single positively rewarded sample (by prompting the model) requires expanding the GRPO batch size to 128. This imbalance between positive and negative training signals impedes RL convergence.

We hypothesize that these issues become more prominent for small language models, where the RL stability is more likely to be fragile, compared to LLMs.

3. **Exploration–Exploitation Tradeoff** Effective exploration is essential for discovering high-reward policies in RL. While a sampling temperature of 1.0 or higher is employed to encourage explorations, a lower temperature (e.g. 0.6) is typically used to constrain output variance on math and coding tasks. In our experiments, we have observed a substantial performance gap resulting from this divergence between the exploration used during training and the exploitation settings applied at evaluation.

To address the aforementioned challenges, we introduce a set of methods to improve the stability and effectiveness of RL training:

1. **Prompt Optimization** We perform multiple rounds of sampling using multiple candidate prompts intended for RL training using the distilled model. Then only those prompts whose generated responses exhibit relatively uniform token lengths are retained. This method mitigates the instability induced by high intra-group response length variance during GRPO optimization.

2. **Reward Rebalancing through Oversampling and Filtering** Inspired by DAPO (Yu et al.), for difficult prompts, we first conduct oversampling to ensure sufficient diversity in the response group. We then re-balance the group by retaining all positive-reward responses and randomly sampling an equal number of negative-reward responses. To further reduce the length variance and avoid instability from overly easy prompts, we filter out prompts whose group-level accuracy exceeds a certain threshold (e.g., 50%).

3. **Temperature Annealing** To seek the best tradeoff between exploration and exploitation during the course of model training, we introduce temperature annealing. We initialize the sampling temperature as 1.0 and linearly decay it over the first 50% of training steps down to 0.6. For the remaining training steps, the temperature is fixed as 0.6. This strategy encourages broader exploration in the early stage of RL while gradually transitioning toward the exploitation in the well-known state-action subspace.

4 Synthetic CoT Data Generation

To support distillation and rollout-based preference learning, we construct a large-scale reasoning dataset composed of LLM-generated synthesized reasoning trajectories. Specifically, we aggregate multiple public datasets—such as Bespoke (Labs, 2025), Openthoughts (OpenThoughts, 2025), and OpenR1-Math (HuggingFace, 2025)—along with several in-house seed datasets. For datasets that already include reasoning trajectories, we directly use the provided annotations. For datasets lacking such trajectories, we retain only the math questions and generate new chain-of-thought answers using DeepSeek-R1 (671B). For each question, we sample approximately eight rollouts. An overview of the data sources is provided

Data Resource	Size	Reasoning
AquaRAT (Ling et al., 2017)	98K	✘
Ape210K (Zhao et al., 2020)	210K	✘
MetaMathQA (Yu et al., 2023)	395K	✘
MathInstruct (Yue et al., 2023)	262K	✘
TAL-SCQ5K (TAL-SCQ5K, 2023)	5K	✘
OpenR1-Math (HuggingFace, 2025)	220K	✔
Bespoke-Stratos-17k (Labs, 2025)	17K	✔
OpenThoughts-114K (OpenThoughts, 2025)	114K	✔

Table 2: Overview of the data resources used for constructing the reasoning dataset. For non-reasoning data, we only use the questions and sample answers from Deepseek R1.

in Table 2. In total, we collect around 10 million rollouts across 1.6 million samples, including contributions from public datasets. For math questions that are verifiable, we first apply a math-verification tool to assess the correctness of the answers. However, as automatic verification can sometimes fail to validate complex solutions—leading to false negatives—we additionally employ GPT-4o-mini to re-verify rollouts initially flagged as incorrect. To maintain dataset balance, we annotate each data sample with attributes including the domain category, the difficulty level, and the presence of repetitive patterns. Domain categories cover a wide range of areas such as algebra, geometry, theory, probability, and calculus. Difficulty levels are categorized as elementary school, middle school, high school, college, and graduate level. The mid-training phase leverages the full dataset, while subsequent training steps operate on selected subsets.

5 Experiment

5.1 Evaluation

We evaluate our model on three mathematical reasoning tasks: AIME24 (MAA, 2024), Math-500 (Lightman et al., 2023), and GPQA Diamond (Rein et al.). For evaluation, the generation parameters are set with a temperature of 0.6, top_p of 0.95, and a maximum sequence length of 32K. For each task, we conduct 3 runs and report the average performance across these trials.

5.2 Baselines

We compare our `Phi-4-Mini-Reasoning` model with o1-mini and several leading open-source, small-scale reasoning models, including DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), Bespoke-Stratos-7B (Labs, 2025), and OpenThinker-7B (OpenThoughts, 2025).

5.3 Training Settings

For the first two distillation stages, we use a batch size of 128, a learning rate of 1e-5, a total of 5 training epochs, and a warmup ratio of 0.1. During the first stage, the sequence length is set to 16K *with* packing strategy, whereas in the second stage the sequence length is extended to 20K *without* packing. For the Rollout DPO phase, we use a learning rate of 5e-7 for a single training epoch, with a sequence length of 16K. During the RL stage, a learning rate of 5e-7 and a sequence length of 25k are used to encourage model exploration.

5.4 Results

The overall results are presented in Table 3. `Phi-4-Mini-Reasoning`, despite having only 3.8 billion parameters, outperforms all open-source baseline models, including those nearly twice its size. In addition, we provide an ablation study to demonstrate the contribution of each training stage to the performance of `Phi-4-Mini-Reasoning`.

Model	AIME	MATH-500	GPQA Diamond
o1-mini*	63.6	90.0	60.0
DeepSeek-R1-Distill-Qwen-7B	53.3	91.4	49.5
DeepSeek-R1-Distill-Llama-8B	43.3	86.9	47.3
Bespoke-Stratos-7B*	20.0	82.0	37.8
OpenThinker-7B*	31.3	83.0	42.4
Llama-3.2-3B-Instruct	6.7	44.4	25.3
`Phi-4-Mini`	10.0	71.8	36.9
+ Distill Mid-training	30.0	82.9	42.6
+ Distill Fine-tuning	43.3	89.3	48.3
+ Roll-Out DPO	50.0	93.6	49.0
+ RL (`Phi-4-Mini-Reasoning`)	**57.5**	**94.6**	**52.0**

Table 3: Pass@1 CoT Reasoning results of `Phi-4-Mini-Reasoning` compared with larger 7B reasoning models and OpenAI models. An asterisk (*) indicates results taken directly from the published reports, while the remaining results were reproduced in our work.

5.5 Ablations

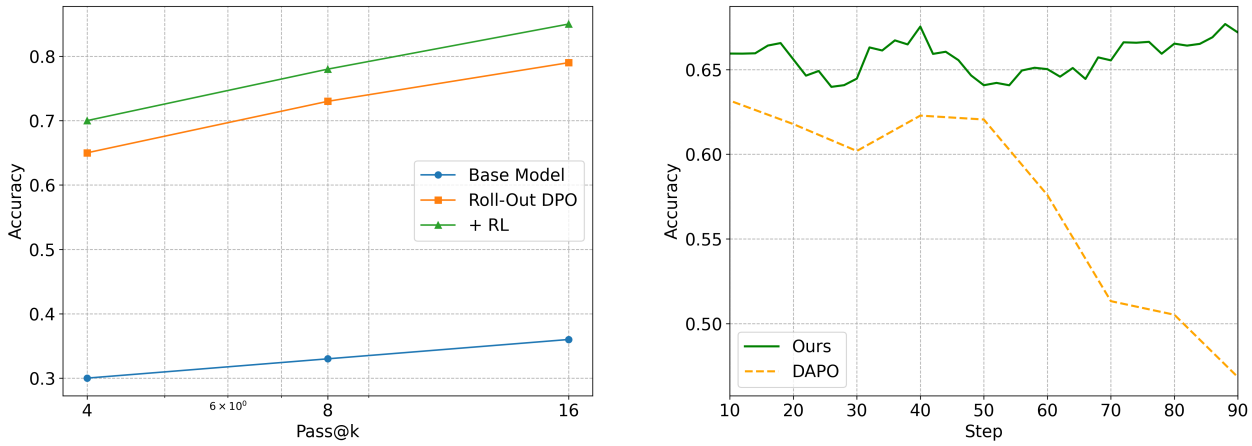
In this section, we conduct ablation studies to understand the impact of our distillation training on the model's reasoning capability and compare the training stability of our RL recipe with DAPO.

To measure the reasoning boundary of an LLM, we use the pass@k metric. For each problem, we sample k outputs from the model. The pass@k value for a question is 1 if at least one of the k samples passes verification; otherwise, it is 0. The average pass@k over the dataset reflects the proportion of problems that the model can solve within k attempts. As shown in Figure 2a, our distillation pipeline serves as an effective approach for injecting reasoning-related knowledge into the model. After the distillation phase, pass@k scores are substantially improved, indicating that distillation successfully extends the reasoning capability boundary of the base LLM. This lays a strong foundation for subsequent RL training. Building on this, RL fine-tuning further improves performance, providing an additional boost of approximately 7 points on average and further refining the model's abilities.

We also compare our RL training method against DAPO. As shown in Figure 2b, DAPO does not perform well in our setting: the consensus@16 metric on the AIME dataset consistently degrades as training progresses. In contrast, our RL training technique exhibits greater stability and consistently yields meaningful improvements over the base model.

5.6 Safety Statement

Phi-4-Mini-Reasoning was developed in accordance with Microsoft's responsible AI principles. Potential safety risks in the model's responses were assessed using the Azure AI Foundry's Risk and Safety Evaluation framework, focusing on harmful content, direct jailbreak, and model groundedness. The



(a) (b)

Figure 2: (a) Pass@k curves on AIME 2024 for the base model, the Roll-Out DPO model, and the model with additional RL training. Rollout DPO significantly improves Pass@k, extending the model's reasoning capabilities. Further RL training yields additional gains. (b) Comparison between DAPO and our RL training method, evaluated by cons@16 accuracy on AIME 2024. Our RL training approach demonstrates better stability.

Phi-4-Mini-Reasoning Model Card contains additional information about our approach to safety and responsible AI considerations that developers should be aware of when using this model.

6 Conclusion

We present a multi-stage training paradigm to enhance reasoning capabilities in small language models (SLMs), combining large-scale distillation, rollout preference learning, and reinforcement learning with verifiable rewards. Applied to Phi-4-Mini, our approach produces `Phi-4-Mini-Reasoning`, a compact 3.8-billion-parameter model that outperforms open-source reasoning models nearly twice its size. We demonstrate that a carefully coordinated sequence of training stages is essential for unlocking robust reasoning in SLMs. Our results show that small models, when trained with deliberate data selection and training strategies, can match or even exceed the capabilities of much larger models. We believe that this work provides a blueprint for developing efficient, high-performing models under resource constraints.

Acknowledgments

We extend our sincere gratitude to Amit Garg, Daniel Perez-Becker, Nguyen Bach, Tetyana Sych and the entire GenAI team for their invaluable contributions to this work. Their support in model review, deployment, and productization was instrumental in bringing this project to completion. We also gratefully acknowledge the Turing team for their ongoing technical collaboration and insightful discussions.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

HuggingFace. 2025. Open r1: A fully open reproduction of deepseek-r1.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Bespoke Labs. 2025. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation. https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation. Accessed: 2025-01-22.

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

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

MAA. 2024. American invitational mathematics examination–aime. In American Invitational Mathematics Examination–AIME 2024.

Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

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

OpenAI. 2024. Learning to reason with llms. *https://openai.com/index/learning-to-reason-with-llms*.

OpenThoughts. 2025. Open Thoughts. https://open-thoughts.ai.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

TAL-SCQ5K. 2023. TAL-SCQ5K: A high-quality mathematical competition dataset in english and chinese. Accessed: 2025-04-26.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024a. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024b. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Guanghao Ye, Khiem Duc Pham, Xinzhi Zhang, Sivakanth Gopi, Baolin Peng, Beibin Li, Janardhan Kulkarni, and Huseyin A Inan. 2025a. On the emergence of thinking in llms i: Searching for the right intuition. *arXiv preprint arXiv:2502.06773*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025b. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, et al. 2025. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.