Skill-Aware Data Selection and Fine-Tuning for Data-Efficient Reasoning Distillation

Lechen Zhang Yunxiang Zhang Wei Hu Lu Wang
University of Michigan, Ann Arbor
{leczhang, yunxiang, vvh, wangluxy}@umich.edu

Abstract

Large reasoning models such as DeepSeek-R1 achieve strong performance on complex reasoning tasks but their size and computational demand limit practical use. Distilling their reasoning capabilities into smaller models via supervised fine-tuning offers a way to democratize reasoning ability, but resource constraints demand data-efficient training strategies. We propose a *skill-centric* distillation framework with two components: (1) **skill-based data selection**, which preferentially samples more examples for skills where the model shows lower proficiency from a large pool of expert reasoning traces, and (2) **skill-aware fine-tuning**, which trains models to explicitly articulate the sequence of skills they will apply before solving a problem, reinforcing skill composition and improving generalization. Operating within a budget of 1,000 training examples, our distillation framework consistently outperforms the standard baseline of fine-tuning with randomly sampled data. Our approach yields average absolute accuracy improvements of +1.6% with Qwen3-4B and +1.4% with Qwen3-8B across five mathematical reasoning benchmarks. Further analysis confirms that these gains are aligned with the emphasized skills, validating the efficacy of targeted training for data-efficient reasoning distillation.

1 Introduction

Large reasoning models such as DeepSeek-R1 [2] have demonstrated impressive performance on complex reasoning tasks, but their size and computational demands make them difficult to use in practice. Distilling these capability into smaller models via supervised fine-tuning (SFT) is a promising way to broaden the access, especially in light of recent findings that high-quality small reasoning data can outperform much larger ones [21]. A key challenge, however, is the strategy of choosing the right SFT data. Current pipelines typically adopt a one-size-fits-all approach, treating all training examples uniformly [2]. This overlooks the latent structure of data—such as the underlying skills and difficulty of examples—as well as the model's current knowledge state. In contrast, human learning is highly structured—specialized training that builds on a learner's existing knowledge often proves most effective [4, 19]. Motivated by this analogy, we ask whether structured train data selection can similarly benefit LLMs, improving both learning efficiency and generalization in long-form reasoning tasks such as mathematics.

In the context of LLM training, a *skill* is usually defined as an "atomic" learned competency (e.g., addition, multiplication, etc.) when solving problems [1, 10]. Recent studies [10] have explored skill-oriented training for LLMs, often by classifying training data at a high level and emphasizing broader skill coverage and data quality as key factors for performance gains [21]. Yet these approaches have two limitations. First, these approaches usually segment problems at the level of whole questions [22], ignoring the fact that a single question often involves many atomic skills (e.g., multiplication skill appears across countless tasks). These overlaps in skills and problems was largely ignored in prior work. Second, prior work [14, 21] generally does not adapt training to the strengths and weaknesses

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

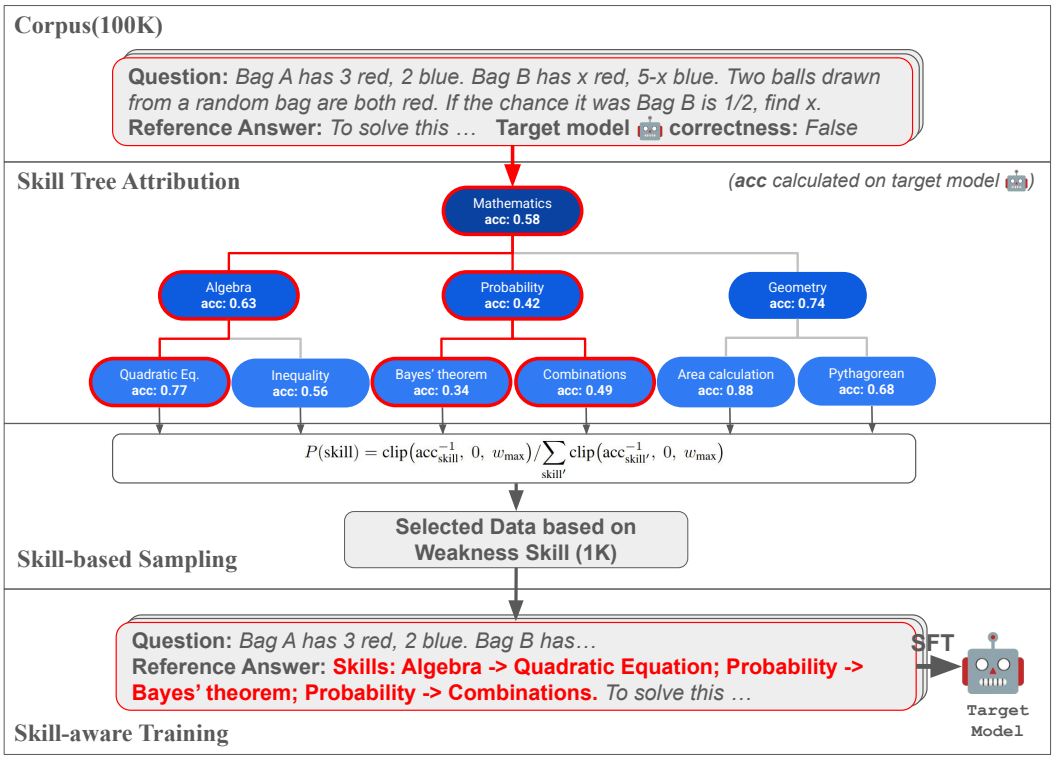

Figure 1: Overview of our skill-centric distillation framework. (1) **Skill Tree Attribution:** Each problem is mapped to nodes on a hierarchical skill tree [9] via top-down LLM-based skill attribution (2) **Skill-based Sampling:** The target model's per-skill accuracy guides sampling, with weaker skills emphasized. (3) **Skill-aware Training:** Selected examples are augmented with explicit skill chains (as shown in red) for skill-aware training.

of specific LLMs. A model already proficient in geometry, for example, should not be saturated with additional geometry training. Our approach rests on a simple principle: LLMs should be trained more intensively on the atomic skills they struggle with, and less on the ones they already master. Explicitly identifying and tracking latent skills—such as arithmetic, factoring, or subgoal planning—can enable more targeted training and diagnosis.

Another challenge lies in enabling LLMs to grasp the hierarchical structure of skills. In typical training, models only see input–output pairs, and the relationships among underlying skills remain implicit. Prior work [3] has shown that prompting models with an explicit list of skills can significantly improve performance. Inspired by this, we inject structured skill information into training data so that models not only learn to solve problems but also internalize how different levels of skills relate to each other.

In this work, we introduce a *skill-centric* data construction framework for training LLMs on mathematical reasoning. The hierarchical structure of the skill tree allows new training examples to be mapped quickly onto multiple relevant skill chains. By estimating a model's proficiency in each skill, we can then select targeted subsets of data for training. Moreover, by embedding interpretable skill chains into the data, the model learns to reason explicitly over a set of skills before attempting to solve a problem. Experiments demonstrate that both skill-based data selection and skill-aware data construction yield clear improvements in model performance, highlighting the promise of structured, skill-aware training for advancing LLM reasoning.

2 Method

Our approach is motivated by two simple intuitions: (1) models should receive more training data on skills they are weak at, and (2) models can generalize more effectively if they are explicitly trained to recognize explicit skill structures. Our workflow, as shown in Figure 1, begins with a curated corpus

of 100K math QA pairs, and a pre-defined skill tree that categorizes mathematical problems into hierarchical skills.

Step 1: Skill tree attribution Each training problem is mapped onto the tree by attributing its reference solution to relevant skills. Starting from the root, we prompt `Qwen/Qwen2.5-32B-Instruct`[15] to decide which high-level skill is involved (prompt shown in Appendix C). For each selected skill, the LLM is further asked to drill down the decision at the next level of the tree, until the leaf node is reached. This recursive process leverages the hierarchical structure (with $O(\log N)$ complexity) to avoid overwhelming the model with a flat multi-label decision and ensures comprehensive coverage of all required skills. [1]

Step 2: Skill-based sampling To adapt training data to a model's weaknesses, we evaluate the target model on the 100K corpus. For each leaf skill, we compute the model's accuracy, yielding a skill-wise performance profile. Training examples are then sampled with probabilities inversely proportional to these accuracies:

$$P(\text{skill}) = \frac{\text{clip}\big(\text{acc}_{\text{skill}}^{-1},\ 0,\ w_{\max}\big)}{\sum_{\text{skill}'} \text{clip}\big(\text{acc}_{\text{skill}'}^{-1},\ 0,\ w_{\max}\big)}$$

where $w_{\max}$ is empirically set to 10,000 to cap divide-by-zero issue. This mechanism ensures that underrepresented or difficult skills are emphasized while preventing excessive redundancy in well-mastered ones. Using this distribution, we construct training subsets of 1K examples.

Step 3: Skill-aware training Finally, we prepare skill-aware variants of the training data by embedding the explicit skill chain into each instance. For each problem, the ordered sequence of required skills—e.g., *"Skills: [Mathematics → Probability → Bayes' theorem]"*—is prepended before the solution. This encourages the model to explicitly traverse the required skills before attempting the solution, rather than relying on shortcuts, enabling fine-grained diagnostics of model performance at the skill level.

3 Experiments

We conduct a series of experiments to evaluate the effectiveness of our skill-tree-based data selection framework for SFT in mathematical reasoning tasks.

3.1 Setup

We experiment with two reasoning models from the Qwen3 family: Qwen3-4B and Qwen3-8B [16]. We use **OpenMathReasoning** [13] as the primary dataset, a large-scale math reasoning corpus containing 306K unique problems with 3.2M solutions sampled from DeepSeek-R1 [2]. From this corpus, we extract a clean set of 100K unique QA pairs as our training pool (Details in Appendix B). We ensure that there is no data leakage between our training corpus and the evaluation benchmarks. In our experiments, we adopt the existing 3-layer skill tree structure proposed in the *Instruct-SkillMix* paper [9] and labeled skills for all data, though our pipeline is readily adaptable to other skill tree designs.

All models are fine-tuned for 5 epochs unless otherwise noted (details in Appendix D). Evaluation is conducted on five diverse math benchmarks: **AMC23**, **AIME2024**, **AIME2025**, **MATH L5** (Level 5) [7], and **OlympiadBench** [6], all consisting of competition-style math questions. Avg@8 accuracy (calculated by the average accuracy over 8 independent samples per question) is reported; however, for random selection, we averaged the Avg@8 over three random seeds.

3.2 Main Results and Analysis

Table 1 shows the performance across different training strategies. We observe that: **Skill-tree-based data selection generally outperforms random sampling.** For Qwen3-4B, Skill-based data selection

[1] We manually inspected ∼100 random QA pairs and found no evidence of missing or mislabeled skills.

Base Model	Data Selection	Fine-tuning Strategy	AMC23	AIME2024	AIME2025	MATH L5	OlympiadBench	Average
Qwen3-4B	-	Base	90.1	61.1	50.7	84.3	49.1	67.1
	Full (100K)	Standard SFT	81.9	46.7	34.6	80.2	47.0	58.1
	Random	Standard SFT	89.5	60.1	50.3	85.3	49.0	66.8
	Random	Skill-aware SFT	90.9	62.2	49.9	**85.8**	49.0	67.6
	Skill-based	Standard SFT	89.1	62.5	50.0	85.5	49.5	67.3
	Skill-based	Skill-aware SFT	**91.9**	**64.6**	**50.8**	85.3	**49.6**	**68.4**
Qwen3-8B	-	Base	88.2	61.1	50.2	84.7	49.1	66.7
	Full (100K)	Standard SFT	82.4	47.1	35.5	80.6	46.7	58.5
	Random	Standard SFT	90.2	62.6	50.8	86.0	50.7	68.1
	Random	Skill-aware SFT	91.5	65.7	**52.6**	**86.6**	50.4	69.4
	Skill-based	Standard SFT	**93.4**	62.1	51.3	86.2	49.7	68.5
	Skill-based	Skill-aware SFT	91.9	**67.1**	50.0	**86.6**	**51.6**	**69.5**

Table 1: Accuracy (%) of Qwen3-4B and Qwen3-8B under different training data selection and fine-tuning strategies using **1K training examples**. Each column of a base model is **bolded** at its highest value and underlined at its second highest. Results are reported using Avg@8 across five math benchmarks.

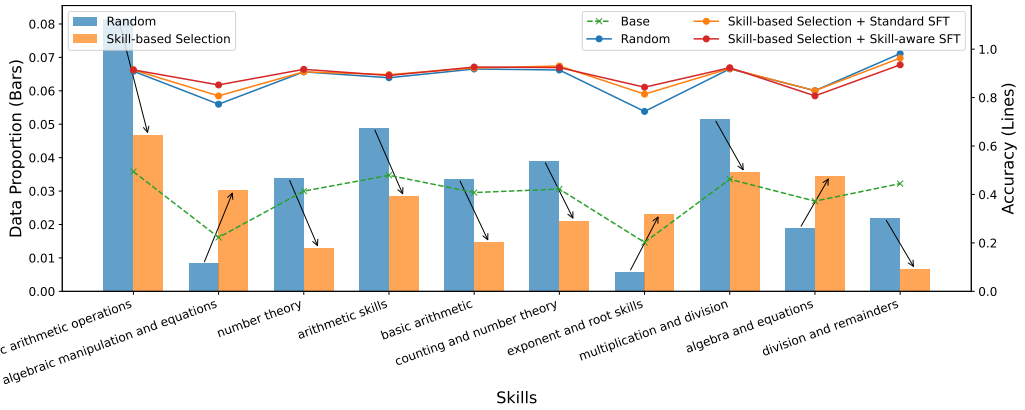

Figure 2: Data proportion shift of skill-based selection and per-skill accuracy on **MATH-500**. Skill-based sampling improves weaker skills while preserving strong ones, flattening the accuracy curve toward balanced mastery. Skill-aware augmentation further enhances robustness across skills.

yields a +0.5 gain in average accuracy, with the largest improvements on AIME2024 (+2.4). Similarly, Qwen3-8B has a +0.4 average gain with significant improvement on AMC23 (+3.2). These results indicate that aligning training with the model's weaker skills provides consistent benefits to LLMs. Second, **skill-aware training consistently provides additional gains.** Adding explicit skill chains improves average accuracy in nearly all settings, with the largest boost on AIME2024 (up to +5.0), and strongest overall gains of up to +0.8 for Qwen3-4B and +1.3 for Qwen3-8B. Combining Skill-based data selection with skill-aware augmentation further amplifies the effect, yielding significant improvements over random selection (+1.6 for Qwen3-4B and +1.4 for Qwen3-8B) and delivering the strongest overall results, including challenging benchmarks such as AIME2024 and OlympiadBench. These findings confirm that skill-aware sampling and training are complementary and robust.

To examine the effect of our skill-based oversampling strategy on individual skills, we further evaluate 500 problems from the **MATH-500** [12] benchmark. As presented in Figure 2, the bars show the distribution shift of sampled data under skill-based sampling, while the lines report skill-wise accuracies across different settings (each position on the x-axis represents a distinct skill). We observe clear improvements over the base model: both random and skill-based sampling substantially improve accuracy over the base model. **Skill-based oversampling effectively aligns finetuning data distribution with model weaknesses.** For weaker skills (those below the average accuracy), skill-based oversampling leads to large improvements that bring performance close to the overall average (e.g., algebraic manipulation and equations). Moreover, the accuracy of stronger skills remains high although sampled less frequently, suggesting that random sampling may waste training cost on areas where the model already performs well. Therefore, **skill-based training curve becomes notably flatter, showing that the model achieves more balanced and robust performance across skills**. Adding skill-aware augmentation further strengthens this effect, yielding even greater consistency in skill performance.

3.3 Ablation Studies

Effect of Sampling Aggressiveness We examine how the aggressiveness of skill-based weakness sampling influences performance by varying the exponent of accuracy (replacing the inverse acc^{-1} in the formula with acc^{-T}). As shown in Table 2, performance slightly improves and then saturates as sampling becomes more aggressive, but significantly degrades when $T < 1.0$. Thus, setting $T = 1.0$ provides a simple and effective balance.

Is the Full Chain of Skills Necessary? Our skill-aware SFT provides models with the full hierarchical skill chain. But is this structure indispensable? To examine this, we test variants that expose only a single layer, either the top-level skills or only the leaf skills. As shown in Table 2, providing only high-level skills yields minimal gains, while using only leaf-level skills leads to moderate improvement but still lags behind the full chain. This suggests that exposing the complete skill tree structure during training is beneficial for model learning.

Setting	Avg Accuracy
Effect of Sampling Aggressiveness	
$T = 0.5$	70.7
$T = 0.75$	71.3
$\mathbf{T = 1.0}$	71.9
$T = 2.0$	72.0
$T = 3.0$	71.9
Is the Full Skill Chain Necessary?	
Full skill chain	72.9
Root Skills Only	72.2
Leaf Skills Only	72.7

Table 2: Ablations on Sampling Aggressiveness and Hierarchical Skill Chain. Default settings are **bolded**. Full results are in Appendix Table 3.

4 Conclusion

This research demonstrates that skill-based data selection and skill-aware training enable more capable, data-efficient, and interpretable reasoning distillation. By prioritizing examples of weaker skills and embedding explicit skill structures during fine-tuning, our approach allows smaller models to acquire more balanced and robust reasoning abilities. These findings highlight the potential of skill-centric training as a general framework for improving training efficiency and transparency.

References

[1] Mayee F Chen, Nicholas Roberts, Kush Bhatia, Jue WANG, Ce Zhang, Frederic Sala, and Christopher Re. Skill-it! a data-driven skills framework for understanding and training language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[2] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[3] Aniket Rajiv Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy P Lillicrap, Danilo Jimenez Rezende, Yoshua Bengio, Michael Curtis Mozer, and Sanjeev Arora. Metacognitive capabilities of LLMs: An exploration in mathematical problem solving. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[4] Robert Gagne. The acquisition of knowledge. *Psychological Review*, 69:355–365, 07 1962.

[5] Shousheng Jia Haosheng Zou, Xiaowei Lv and Xiangzheng Zhang. 360-llama-factory, 2024.

[6] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[7] 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. *NeurIPS*, 2021.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[9] Simran Kaur, Simon Park, Anirudh Goyal, and Sanjeev Arora. Instruct-skillmix: A powerful pipeline for LLM instruction tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Jiazheng Li, Lu Yu, Qing Cui, Zhiqiang Zhang, JUN ZHOU, Yanfang Ye, and Chuxu Zhang. MASS: Mathematical data selection via skill graphs for pretraining large language models. In *Forty-second International Conference on Machine Learning*, 2025.

[11] Yang Li, Youssef Emad, Karthik Padthe, Jack Lanchantin, Weizhe Yuan, Thao Nguyen, Jason Weston, Shang-Wen Li, Dong Wang, Ilia Kulikov, and Xian Li. Naturalthoughts: Selecting and distilling reasoning traces for general reasoning tasks, 2025.

[12] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[13] Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

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

[15] Qwen Team. Qwen2.5: A party of foundation models, 2024.

[16] Qwen Team. Qwen3: Think Deeper, Act Faster | Qwen, April 2025.

[17] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[18] Lin Sun, Guangxiang Zhao, Xiaoqi Jian, Yuhan Wu, Weihong Lin, Yongfu Zhu, Change Jia, Linglin Zhang, Jinzhu Wu, Junfeng Ran, Sai er Hu, Zihan Jiang, Junting Zhou, Wenrui Liu, Bin Cui, Tong Yang, and Xiangzheng Zhang. Tinyr1-32b-preview: Boosting accuracy with branch-merge distillation, 2025.

[19] Richard T. White. Research into learning hierarchies. *Review of Educational Research*, 43(3):361–375, 1973.

[20] Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. Table-r1: Inference-time scaling for table reasoning, 2025.

[21] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *Second Conference on Language Modeling*, 2025.

[22] Zhiyuan Zeng, Yizhong Wang, Hannaneh Hajishirzi, and Pang Wei Koh. Evaltree: Profiling language model weaknesses via hierarchical capability trees. In *Second Conference on Language Modeling*, 2025.

[23] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics.

A Related Work

Distillation of LLM Reasoning Knowledge distillation was first introduced as a way to compress large neural networks into smaller ones [8], later popularized in NLP [17]. Building on these foundations, recent work has shifted toward distilling reasoning abilities. For example, **(author?)** [2] showed that large expert models can successfully transfer complex multi-step reasoning traces into smaller students with ~800k R1 outputs, establishing a practical recipe for democratizing reasoning power. Follow-up studies such as OpenMath-Nemotron [13], Table-R1 [20], Tiny-R1 [18] demonstrate that distilling reasoning can yield compact models that approach or even rival much larger systems in reasoning capability.

Data Selection for Efficient Training Selecting the most informative training examples has long been studied as a way to improve model performance under limited data budgets. Recent studies show that small but carefully curated datasets can yield strong reasoning performance. For example, LIMO [21], s1 [14], and NaturalThoughts [11] all demonstrate that high-quality and diverse examples often outperform large-scale random sampling. Building on this insight, structured selection methods such as MASS [10] and Skill-It [1] leverage graphs and learning dependencies to guide sampling, enhancing efficiency by prioritizing the most instructive reasoning examples.

Skill Decomposition and Structured Reasoning A growing line of work views complex reasoning as a composition of simpler skills and leverages this structure for improved evaluation and training. Didolkar et al. [3] showed that prompting LLMs to identify relevant skills improves math performance. Zeng et al. [22] introduced EvalTree, which organizes tasks into a hierarchical skill tree to locate weak skills for synthesizing targeted data. Instruct-SkillMix [9] combines pre-defined skills to create instruction data, enabling an 8B model to match far larger ones.

B Data Filtering Details

From the **OpenMathReasoning** corpus [13], we construct an 100K clean training pool by applying several filtering steps. First, we discard problems without a ground-truth answer. Each unique problem is associated with approximately ten candidate responses; we retain only those generated by DeepSeek-R1 [2] and only when the predicted final answer exactly matches the ground truth. For each problem, we then keep a single valid response to avoid duplication. This procedure yields roughly 105K problem–solution pairs. Finally, we randomly remove 5K instances to obtain a balanced set of 100K unique QA pairs used in our experiments.

C Skill Tree Attribution Details

The prompt we used on `Qwen/Qwen2.5-32B-Instruct`[15] for top-down skill attribution are listed below:

```
Given the following Math problem:

Q&A: {qa_input}

Which of the following skills are involved to understanding or
    solving the problem? Even the most basic skills such as simple
```

```
    addition and subtraction must be taken into account. You can
    select multiple options if needed. Just return a list of skill
    names.

Skills:
{chr(10).join([f"- {name}" for name in child_names])}

Answer as a Python list of strings.
'''
```

D Training Details

Environment. All experiments were conducted using NVIDIA A40 GPUs with 48GB memory. The software environment was configured as follows:

- `360-LLaMA-Factory` [5] (A long-CoT adapted version of `LLaMA-Factory` 0.9.1 [23])
- `torch` 2.7.0
- `transformers` 4.51.3
- `accelerate` 1.0.1
- `datasets` 3.1.0
- `trl` 0.9.6
- `peft` 0.12.0
- `deepspeed` 0.14.4

SFT Training. For SFT training, we used the following settings:

- Batch size: 32 (8 GPUs * 4 Gradient Accumulation)
- Epoch: 5
- Learning rate: 1e-5
- Optimizer: AdamW
- Learning rate scheduler: cosine with warmup
- Warmup ratio: 0.1
- Cutoff length: 8192
- Time Cost: 4 hours per run

Decoding Setup. During inference, we applied the following decoding settings:

- Temperature: 0.6
- Max tokens: 16384
- Top-p: 0.95

E Additional Experiment Results

Full version of Table 2 is shown in Table 3.

Ablation	Setting	AMC23	AIME2024	AIME2025	MATH L5	Average
Effect of Sampling Aggressiveness	$T = 0.5$	89.7	60.0	47.9	85.2	70.7
	$T = 1.0$	89.1	**62.5**	**50.0**	<u>85.7</u>	<u>71.9</u>
	$T = 2.0$	<u>90.6</u>	**62.5**	<u>48.8</u>	**85.9**	**72.0**
	$T = 3.0$	**91.6**	<u>61.7</u>	<u>48.8</u>	85.6	<u>71.9</u>
Is the Full Skill Chain Necessary?	Full skill chain	**91.9**	**64.2**	<u>50.4</u>	85.1	**72.9**
	Root Skills Only	<u>91.6</u>	58.3	**52.5**	<u>86.3</u>	72.2
	Leaf Skills Only	90.9	<u>62.9</u>	50.0	**86.9**	<u>72.7</u>

Table 3: Ablations on sampling aggressiveness (T) and on exposing different portions of the skill hierarchy during skill-aware SFT. Within each ablation block, the highest value per column is **bolded** and the second-highest is <u>underlined</u>.