Report on Smaller, Weaker, Yet Better: Training LLM Reasoners via Compute-Optimal Sampling

Seminar Machine Learning, Winter Term 2024/2025

Faidra Anastasia Patsatzi

December 11, 2024

Abstract

The paper ”Smaller, Weaker, Yet Better: Training LLM Reasoners via Compute-Optimal Sampling” [Ban+24] by Google DeepMind has been accepted at the NeurIPS 2024 MATH-AI Workshop. This report aims to summarize the paper’s main contributions, provide insights into the most important arguments, and comment on the results, related work, and future directions.

1 Introduction

Techniques used in training of language models (LMs) have received a lot of attention by the research community in recent years, with multiple works on training trends of LMs, such as scaling laws to empirically model the trend of increasing model performance as a function of the model size [Kap+20] and of the available training compute as well [Hof+22]. In the finetuning setting, practices where synthetic data (i.e., data generated by an LM) is used instead of human-labeled data are gaining popularity, since they are less expensive, more efficient, and lead to better performance [Sin+24]. One example is knowledge distillation, where a student (typically smaller) LM is finetuned using data generated by a teacher LM [HVD15]. Intuitively, often larger LMs are used for data generation to guarantee high quality of the synthetic data [Muk+23]. A popular practice is using an LLM to sample solutions for each question multiple times, keeping only the ones where the final answer is correct, and using the correct solutions (among the synthetic ones) in finetuning [Zel+22]. In practice, this approach has the limitation that sampling strong LMs for synthetic data can be expensive and compute-intensive, therefore the amount of synthetic data generated realistically depends on a given fixed sampling budget.

In the discussed paper [Ban+24], the authors argue that, under a given fixed compute budget, using a weaker and cheaper (WC) LM is preferable to using a stronger but more expensive (SE) LM for finetuning a target LM on reasoning tasks. They demonstrate that 1) the WC-generated data reach higher diversity and coverage scores, but also a higher false positive rate, and 2) models finetuned on WC synthetic data for reasoning outperform models finetuned on SE synthetic data in a compute-matched comparison, i.e., finetuning with WC synthetic data can consistently be compute optimal.

2 Methods

Preliminary definitions and methodology from [Ban+24] for synthetic data generation and finetuning are presented in this section.

2.1 Synthetic Data for Reasoning Tasks

Given a training dataset $\mathcal{D} = \{q_i, a_i\}_{i=1}^{i=n}$, where q_i are reasoning problems and a_i are the corresponding final answers, a synthetic dataset appropriate for finetuning LMs on reasoning tasks can be acquired by following the STaR approach [Zel+22]:

1. Generating multiple answers for each question q_i with non-zero temperature, which yields the dataset $\mathcal{D}_G = \{q_i, \{(\hat{r}_{ij}, \hat{a}_{ij})_{j=1}^{j=k}\}\}$ consisting of k samples for each question q_i, where each pair $(\hat{r}_{ij}, \hat{a}_{ij})$ stands for the j-th generated reasoning chain and j-th generated final answer.
2. Removing from $\mathcal{D}_G$ all $(\hat{r}_{ij}, \hat{a}_{ij})$ pairs where $\hat{a}_{ij}$ does not match the ground truth answer a_i.
3. Using the filtered data $\tilde{\mathcal{D}}_G$ (after step 2.) to finetune a model (details in Appendix A) by maximizing the objective
$$\mathbb{E}_{(q,r,a)\sim\tilde{\mathcal{D}}_G}\left[\log(p_\theta(r, a|q))\right]. \tag{1}$$

The synthetic dataset is evaluated across three criteria: coverage, diversity, and false positive rate (FPR). Let k be the number of samples generated for each question. **Coverage** (coverage@k) is defined as $\mathbb{E}_{\mathcal{D}_G}\left[1 - \binom{M-c}{k}/\binom{M}{k}\right]$, where c is the number of solution pairs $(\hat{r}_{ij}, \hat{a}_{ij})$ with correct answers, out of M solution pairs. The coverage score represents the amount of unique questions with one or more correct solutions when sampling k solution pairs per question. **Diversity** (diversity@k) is calculated as the number of unique correct solutions per problem on average. **False Positive Rate (FPR)** measures the percentage of solution pairs in $\tilde{\mathcal{D}}_G$ where the answer is correct but the reasoning is wrong.

2.2 Compute-Matched Setting for Sampling

Previous studies showing that performance of finetuning using synthetic data scales with the sampled model's size [Sin+24] proceed with a number-matched comparison in their experiments, i.e. they sample the same number of solutions from each model regardless of size, leading to higher compute usage for bigger models. The authors [Ban+24] argue that for a given fixed sampling budget (measured in FLOPs), one can sample either more solutions from a weaker but cheaper model (WC) or fewer solutions from a stronger but more expensive model (SE). Let P_{WC} and P_{SE} denote the number of parameters of the WC and SE models, respectively. Considering decoder-only transformers and that a forward pass accounts for $2P$ FLOPs per token given a model with P parameters [Kap+20], X tokens would require $2P \cdot X$ FLOPs for a forward pass. Consequently, assuming that W inference tokens (inference equals to 1 forward pass) are needed on average for generating each solution, per generated solution $2P \cdot W$ FLOPs would be required. Let S_{WC} and S_{SE} represent the number of solutions sampled per question for the WC and SE models, respectively. Then, the total costs of synthetic data generation can be defined as

$$C_{WC} = n \cdot S_{WC} \cdot W \cdot 2P_{WC} \text{ and } C_{SE} = n \cdot S_{SE} \cdot W \cdot 2P_{SE}\,. \tag{2}$$

In the compute-matched setting, there is a common, fixed sampling budget

$$C = C_{WC} = C_{SE}\,. \tag{3}$$

By substitution of the terms in Eq. 3 with the expressions from Eq. 2 and solving for S_{WC}, we have

$$S_{WC} = \frac{P_{SE}}{P_{WC}} \cdot S_{SE}\,, \tag{4}$$

where the ratio P_{SE}/P_{WC} controls how many more samples can be generated from the WC model under the same compute budget. Note that the compute-matched setting here only concerns the sampling process and not the finetuning of LMs with synthetic data, where the WC-sampled synthetic dataset might be much larger in size than the SE-sampled synthetic dataset.

Table 1: Finetuning setups and corresponding finetuning paradigm. Table adapted from [Ban+24].

Data (↓), Setup (→)	**Student-LM**	**WC-LM**	**SE-LM**
WC	Knowledge distillation	Self-improvement	W2S improvement
SE	Knowledge distillation	Knowledge distillation	Self-improvement

After acquiring the synthetic data, models can be trained on them for a constant number of epochs and their performance after finetuning on either WC or SE data can be evaluated to derive conclusions about the utility of WC versus SE generated data. The authors consider three finetuning paradigms: 1) **Knowledge Distillation** [HVD15] where a student (smaller) LM is finetuned using data generated by a teacher model, 2) **Self-improvement** concerning a LM that is finetuned on samples generated by itself [Hua+22], and 3) **Weak-to-strong-improvement** (W2S-I), introduced by the authors, where the reasoning skills of a strong LM are improved via training on synthetic samples from a weaker LM. Furthermore, the authors present three finetuning setups (target model to finetune): **Student-LM**, **WC-LM**, and **SE-LM**. Table 1 maps the corresponding finetuning paradigm to each setup for two dataset options: 1) WC generated data and 2) SE generated data.

3 Experiments

3.1 Datasets, Models, and Evaluation

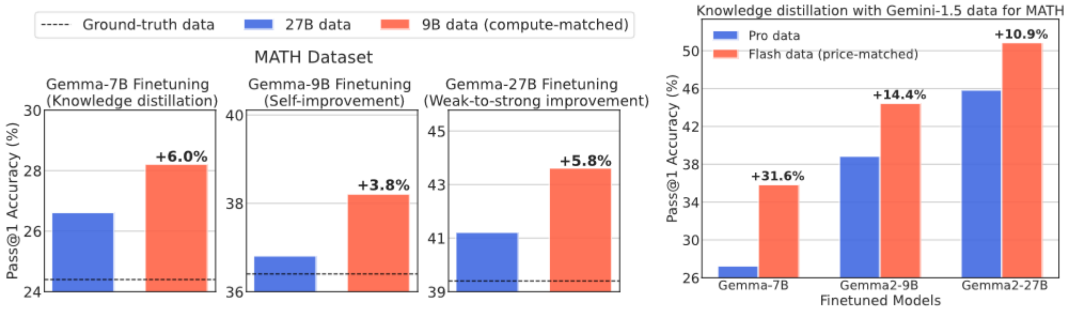
MATH and GSM-8K are used in the experiments as task-specific datasets for reasoning (details in Appendix B). MATH contains competition-level questions covering a range of difficulty levels, whereas GSM-8K consists of elementary school-level math problems. For synthetic data generation, pretrained models Gemma2-9B and Gemma2-27B are used as the WC and SE models, respectively. Two sampling budgets are defined: a low budget, where 1 and 3 samples are generated per question from the WC and SE models, and a high budget, where 10 and 30 samples are generated per question from the WC and SE models. In line with the setups shown in Table 1, Gemma-7B is finetuned as the Student-LM, Gemma2-9B is finetuned as the WC-LM, and Gemma2-27B is finetuned as the SE-LM.

The quality of the synthetic data is measured using the three metrics coverage, diversity, and FPR, as defined in Section 2.1. For the FPR, automatic computation is not possible, since ground truth reasoning chains are not available in the dataset. The authors compute it in two ways: 1) via human evaluation on a smaller part of the data, and 2) via asking Gemini-Pro-1.5 to assess the correctness of the generated reasoning chains. They find that in both cases the FPR estimates are similar. Finetuned models are evaluated via pass@1 accuracy in a zero-shot sampling manner by comparing the generated final answer to the ground truth final answer and calculating the percentage of questions where they match.

3.2 Results and Discussion

The **coverage** score is **higher** for the **WC**-generated data at both low and high sampling budgets and for both datasets, which indicates that generating more samples per question (WC data) enables finding solutions to more unique questions from both MATH and GSM-8K. The **diversity** scores of the **WC** synthetic data are **much higher** than those of the SE synthetic data, showing that more unique reasoning paths from the WC data arrive at correct final answers. However, the **FPR** for **WC** data is **slightly higher** according to the human evaluations, i.e., solutions from the WC model include more often incorrect reasoning chains that lead to correct final answers. The absolute FPR for GSM-8K data is much lower than for MATH data, which can be attributed to the higher difficulty of the problems contained in MATH. The results are visualized in Figures 3 and 4 of Appendix G.

As also shown in Figure 1 for the low sampling budget, in the **Student-LM** setup, Gemma-7B after finetuning with **WC data performed better than with SE data** at both sampling budgets, with higher relative performance gains for the MATH dataset than for the GSM-8K dataset (Figure 5, Appendix G). In the **WC-LM** setup, finetuning with **self-generated data** (WC data) **outperformed** knowledge distillation using **SE** data for both sampling budgets. In **SE-LM** finetuning, the model finetuned with WC-generated data achieved better performance than SE-generated data at both sampling budgets and for both datasets, again with higher relative performance gains for MATH. This is a central observation in the paper and provides solid evidence for the argument that **finetuning a model in the W2S-I setup with WC synthetic data is more compute-optimal than finetuning it via self-improvement on self-generated data (SE data)**. To assess the generalization capabilities of the finetuned models, they are tested with the Functional MATH dataset and the results indicate that WC data further improves generalization over SE data. The main experiments are followed by a range of ablation studies, with their contents and results explained in Appendix D.



(a) Finetuning LMs with Gemma2 data.

(b) Finetuning LMs with Gemini 1.5 data.

Figure 1: Results for models finetuned with MATH synthetic data from (a) Gemma2-9B and Gemma2-27B, and (b) Gemini-1.5-Flash and Gemini-1.5-Pro [Ban+24].

To investigate whether the WC-data finetuned models learn incorrect reasoning chains that lead to correct answers due to the high FPR of the WC synthetic data, the authors measure the FPR of the finetuned models. They find that **finetuned models have similar FPR scores**, regardless of which (WC or SE) synthetic data they have been trained on, suggesting that the higher FPR of the WC data (compared to the FPR of the SE data) does not transfer to a similarly high FPR score difference between the models finetuned with WC vs. SE data.

To demonstrate the generalization of their experimental results, the authors run additional experiments with SoTA LMs, using Gemini-1.5-Pro and Gemini-1.5-Flash as the SE and WC models. Since the models are called via API and their parameter count is unknown, price-matched sampling (based on the price per $1,000,000$ output tokens) is chosen as a proxy to compute-matched sampling. For the two models mentioned above, price-matched sampling corresponds to sampling 1 and 35 solutions per question from the SE and WC models, respectively. In agreement with computed-matched Gemma model experimental results, and as seen in Figure 1, **finetuning with WC data from SoTA model** Gemini-1.5-Flash is found to be **price-optimal**, both in the price-matched and in a more economical setting at 0.15x of the cost (sampling 5 instead of 35 solutions per problem from the WC model).

Finally, the authors also conduct experiments for synthetic data generation where the seed dataset does not include ground-truth labels (final answers). The setup, details, and results are described in Appendix E. Experiments for additional reasoning tasks are described in F.

4 Reflection

One key configuration in the paper is compute-matched sampling. Notably, as defined in Eq. 2, the number of questions n from the dataset is the same and fixed for both WC and SE models. The compute-matched concept here concerns the number of candidate solutions (S_{WC} and S_{SE}) sampled for each question in the dataset. According to the coverage trends provided in Figure 6, Appendix G, it is evident that for datasets such as MATH, that contain difficult problems, coverage stably increases and a much higher coverage score is observed for WC vs. SE compute-matched sampling for the high sampling budget (30 WC solutions vs 10 SE solutions per problem). This is consistent with results from [Bro+24; Son+24], showing that repeated sampling of small models can lead to higher solve rates. However, the coverage trend for GSM-8K appears to converge to high coverage values for both WC number-matched and WC/SE compute-matched sampling (Figure 6, Appendix G). Similarly, for the coding task experiments in Appendix B of the paper, the WC and SE models again reach similar coverage values in the high sampling budget. Therefore, for tasks easier than MATH such as GSM-8K or coding (e.g., MBPP), an interesting direction in the high sampling budget scenario would be trading off the number of WC solutions S_{WC} for more questions n_{WC} from a given dataset. This would, in its simplest form, translate to fixing the number of sampled solutions S per question (e.g., $k = 10$ as in the high sampling budget) and varying the number of questions n_{SE} and n_{WC} with $n_{WC} > n_{SE}$ considered from the dataset, assuming that the dataset is large enough and the given compute budget only allows $n_{SE} < n_{\mathcal{D}}$. If that requirement is not fulfilled (original dataset exhausted), one could consider augmenting it via synthetic question-answer pair generation by taking questions from the original dataset as seeds [Yu+24]. Also experimenting with higher compute budgets would shed more light onto the coverage trend for both datasets and performance of the finetuned models in the higher-budget regime. Moreover, a more rigorous exploration of mixing WC and SE data in future work could point towards different, potentially task-dependent compute-optimal configurations than only using WC data.

Another future direction would be integrating process-based verification instead of only outcome-based verification into the experiments. The finetuned model's FPR assessment was performed as a sanity check in this paper, as mentioned in Section 3.2, to ensure that finetuned models are not significantly affected by the high FPR scores of the synthetic WC data. Process-based verification, e.g., by using an LM verifier, would however ensure that the final synthetic dataset mostly contains samples where not only the final answer is correct, but also the reasoning chain leading to it, ultimately sinking the FPR of WC/SE synthetic data to a much smaller value. A parallel direction, also noted by the authors, would be training LM verifiers for reasoning, e.g., a process-supervised reward model such as in ([Lig+23]), by using synthetic data instead of human-labeled data.

Moreover, the authors also extend their experiments to SoTA models (Gemini family), which they use for synthetic data generation, however the finetuned models are Gemma models in all experiments. To further strengthen their generalization argument, the authors could have also finetuned different open-source pretrained models of similar sizes and capabilities to Gemma-7B/Gemma2-9B/-27B (after pretraining), such as Llama 3/3.1.

Overall, the paper "Smaller, Weaker, Yet Better: Training LLM Reasoners via Compute-Optimal Sampling" [Ban+24], uncovers the potential of using small LMs for synthetic data generation to finetune smaller or larger LMs on reasoning tasks, such as MATH and GMS-8K. The results suggest that using a WC model instead of a SE model for synthetic data is often compute-optimal and encourage a shift away from the common practice of using synthetic data from larger (SE) models to train LM reasoners. In light of the scaling trend of small LMs performance in reasoning (Appendix C), this finding will potentially gain relevance for training LMs on reasoning tasks in the future.

References

[Ban+24] Hritik Bansal et al. *Smaller, Weaker, Yet Better: Training LLM Reasoners via Compute-Optimal Sampling.* 2024. arXiv: 2408.16737 [cs.CL]. URL: https://arxiv.org/abs/2408.16737.

[Bro+24] Bradley Brown et al. *Large Language Monkeys: Scaling Inference Compute with Repeated Sampling.* 2024. arXiv: 2407.21787 [cs.LG]. URL: https://arxiv.org/abs/2407.21787.

[Hof+22] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models.* 2022. arXiv: 2203.15556 [cs.CL]. URL: https://arxiv.org/abs/2203.15556.

[Hos+24] Arian Hosseini et al. *Not All LLM Reasoners Are Created Equal.* 2024. arXiv: 2410.01748 [cs.LG]. URL: https://arxiv.org/abs/2410.01748.

[Hua+22] Jiaxin Huang et al. *Large Language Models Can Self-Improve.* 2022. arXiv: 2210.11610 [cs.CL]. URL: https://arxiv.org/abs/2210.11610.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network.* 2015. arXiv: 1503.02531 [stat.ML]. URL: https://arxiv.org/abs/1503.02531.

[Kap+20] Jared Kaplan et al. *Scaling Laws for Neural Language Models.* 2020. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

[Lig+23] Hunter Lightman et al. *Let's Verify Step by Step.* 2023. arXiv: 2305.20050 [cs.LG]. URL: https://arxiv.org/abs/2305.20050.

[Muk+23] Subhabrata Mukherjee et al. *Orca: Progressive Learning from Complex Explanation Traces of GPT-4.* 2023. arXiv: 2306.02707 [cs.CL]. URL: https://arxiv.org/abs/2306.02707.

[Ouy+22] Long Ouyang et al. *Training language models to follow instructions with human feedback.* 2022. arXiv: 2203.02155 [cs.CL]. URL: https://arxiv.org/abs/2203.02155.

[Sin+24] Avi Singh et al. *Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models.* 2024. arXiv: 2312.06585 [cs.LG]. URL: https://arxiv.org/abs/2312.06585.

[Son+24] Yifan Song et al. *The Good, The Bad, and The Greedy: Evaluation of LLMs Should Not Ignore Non-Determinism.* 2024. arXiv: 2407.10457 [cs.CL]. URL: https://arxiv.org/abs/2407.10457.

[Yu+24] Longhui Yu et al. *MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models.* 2024. arXiv: 2309.12284 [cs.CL]. URL: https://arxiv.org/abs/2309.12284.

[Zel+22] Eric Zelikman et al. *STaR: Bootstrapping Reasoning With Reasoning.* 2022. arXiv: 2203.14465 [cs.LG]. URL: https://arxiv.org/abs/2203.14465.

[Zhe+23] Lianmin Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* 2023. arXiv: 2306.05685 [cs.CL]. URL: https://arxiv.org/abs/2306.05685.

Appendix

A Supervised Finetuning

For supervised finetuning of LMs, the authors maximize the objective from Equation 1, which corresponds to maximizing the probability of of reasoning chain r and final answer a given a question q, where q, r, a are drawn from the filtered dataset $\tilde{\mathcal{D}}_G$. No further details are disclosed about the finetuning process. The authors hint that sequential supervised finetuning is used, presumably following the SFT convention from InstructGPT [Ouy+22], where for each instruction-answer pair (assuming the task is instruction-following) the instruction is provided in the prompt and the model is trained to generate the answer to this instruction (next-token prediction only for the answer).

As a potential future direction, the authors mention experimenting with different finetuning methods, such as iterative finetuning [Sin+24], where after training on a given question-answer pair, the model weights are updated and the updated model is used in the next iteration and trained on the next question-answer pair.

B Datasets

The MATH and GSM-8K datasets are used in the main experiments a task-specific datasets for mathematical problem solving. For MATH, the training, validation, and test sets consist of 7,500, 500, and 500 problems, respectively. For GSM-8K, the training, validation, and test sets consist of 7,500, 500, and 1,319 problems, respectively. In further experiments ran for coding tasks, MBPP (training set: 324 problems, validation set: 100 problems) was used for training, and HumanEval (164 problems) for evaluation. In a further experiment ran for instruction-following, OpenAssistant1 (5,000 instruction-answer pairs) was used for training and IFEval was used for evaluation.

C Trends for Reasoning Models

The findings from the paper shift the focus of synthetic reasoning data generation from larger models towards smaller models. As shown in the Figure 2, the trend of smaller model reasoning capabilities over time grows faster than the one for larger models. As time progresses and model capabilities scale further, it is likely that the results from this paper in favor of using smaller LMs for synthetic data generation might be of even higher relevance in the future. However, in parallel work by Google DeepMind [Hos+24], it is shown that for smaller and cost-efficient models, there exists a gap between LM reasoning capabilities on GSM-8K (standard benchmark) vs. on compositional GSM. Moreover, [Hos+24] argue that smaller models' reasoning process is systematically different, and that long finetuning on GSM-8K human or synthetic data leads to task-specific and benchmark overfitting. These observations should be taken into consideration in future works on training LLM reasoners.

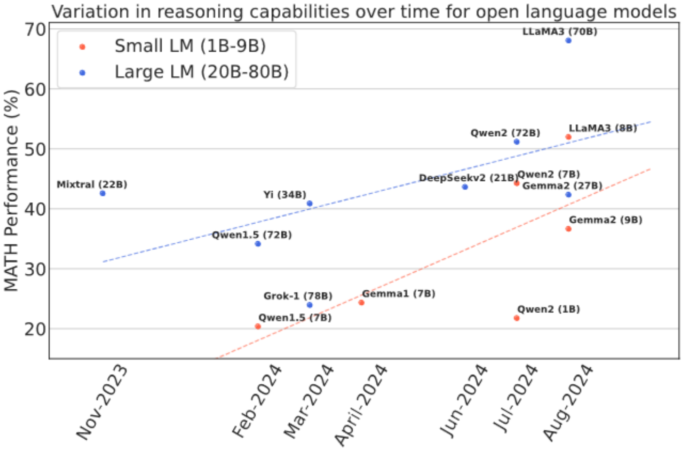


Figure 2: Reasoning skills trend for large and small language models [Ban+24].

D Ablation studies

After the main experiments, the authors conduct a series of ablation studies. They observe similar results (in favor of the WC over the SE synthetic data) when using a much smaller dataset and conclude that finetuning with WC data is beneficial for different dataset sizes. Moreover, they compare finetuning with WC data with the (coverage, diversity) scores (high, high), (high, low), and (low, low), and find that **high coverage and high diversity WC** synthetic data are **better for finetuning** models on reasoning tasks. In a different ablation study, they explore mixing WC with SE data for finetuning, taking a mix of 5 SE and 15 WC solutions per problem, which yields slightly better performance than only using WC data in the WC-LM setup, but worse in the other setups.

E Experiments without ground truth labels

Two scenarios without ground-truth labels are examined: 1) MATH dataset without ground-truth final answers, and 2) instruction-tuning data, where the concept of ground-truth labels doesn't apply. In the first scenario, two strategies are tested: 1a) no verification of the generated solutions, and 1b) verification using LM as a judge [Zhe+23]. The experiments involve the WC and SE models from both the compute-matched and price-matched setups. The results show that more than 65% of Gemma-generated solutions and almost half of Gemini-generated solutions include wrong final answers. In both cases, LM as a judge verification heavily decreases the number of incorrect solutions. Interestingly, finetuning with the Gemma-SE-data outperforms the Gemma-WC-data, whereas finetuning with Gemini-WC-data outperforms the Gemini-SE-data, for both verification setups, suggesting that **finetuning performance for reasoning, without ground-truth data, is highly dependent on the sampled model's quality**.

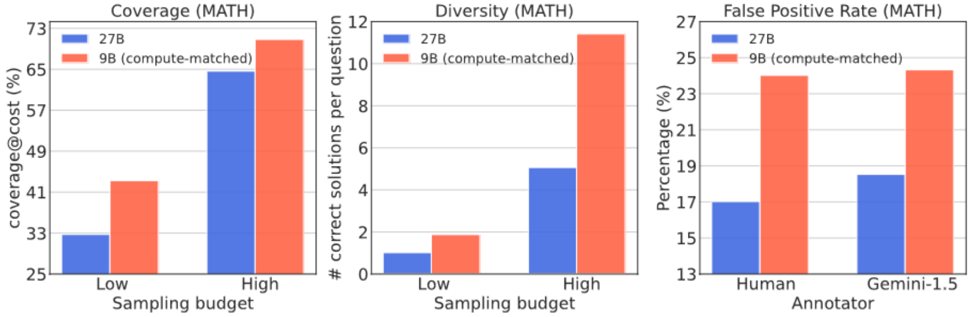
In the second scenario (**instruction finetuning**), models from the Gemini family are used as the SE and WC models (price-matched setting) and are prompted with OpenAssistant1 instructions. Gemma-family models are finetuned on the synthetic data and the **WC data achieve much higher performance** than the SE data for various model sizes (Gemma 7B/9B/27B).

F Additional reasoning tasks

In the experiment for instruction-following tasks, synthetic solutions are generated for the OpenAssistant1 dataset, and models finetuned on the synthetic data are evaluated on IFEval. Only models from the Gemini family are used for synthetic data generation, since they are already instruction-tuned, as opposed to the Gemma models used in the main experiments for mathematic problem solving.

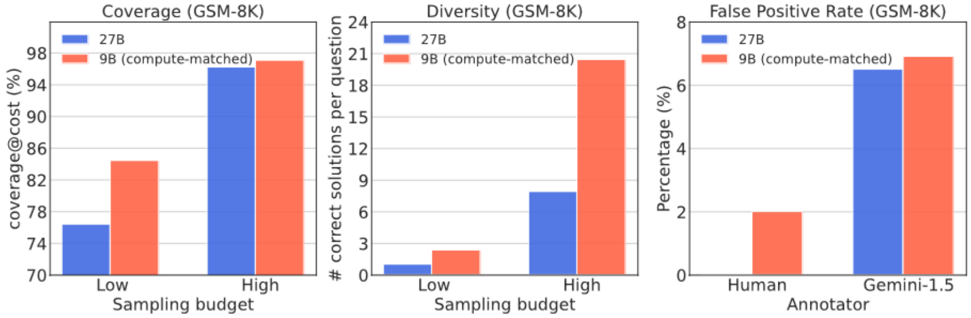
Additional experiments were performed for a coding task (Appendix B of the paper), using the MBPP dataset to generate synthetic solution and HumanEval to evaluate the models finetuned on the synthetic data. In this experiment, Gemma2-9B is used as the WC model and Gemma2-27B as the SE model. At the high sampling budget, both WC and SE data reach similar coverage values, whereas for the low sampling budget, WC data achieves higher coverage. The diversity of WC data is consistently higher. However, results for the finetuned models' performance are mixed: at the low sampling budget, finetuning with WC data yields better performance for the Student-LM and the WC-LM, and similar performance to SE data for the SE-LM. At the high sampling budget, though, SE data outperforms for finetuning the Student-LM, whereas WC data outperforms for finetuning the WC-LM. These findings indicate that **WC compute-optimality might be task-dependent**.

G Additional figures



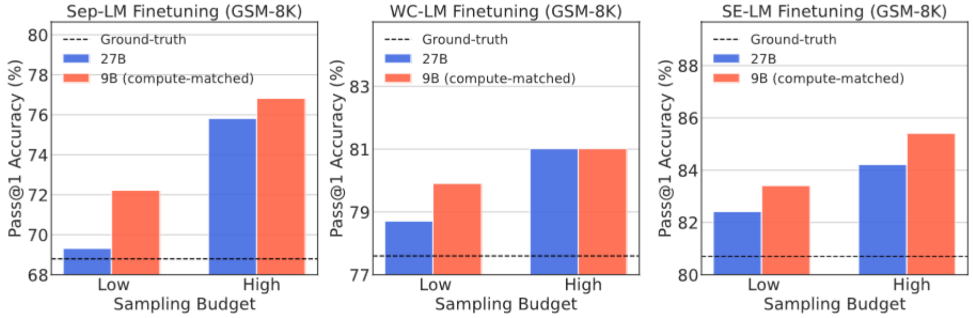
(a) Coverage on MATH. (b) Diversity on MATH. (c) False Positive Rate on MATH.

Figure 3: Diversity, coverage, and false positive rate for the MATH synthetic data [Ban+24].



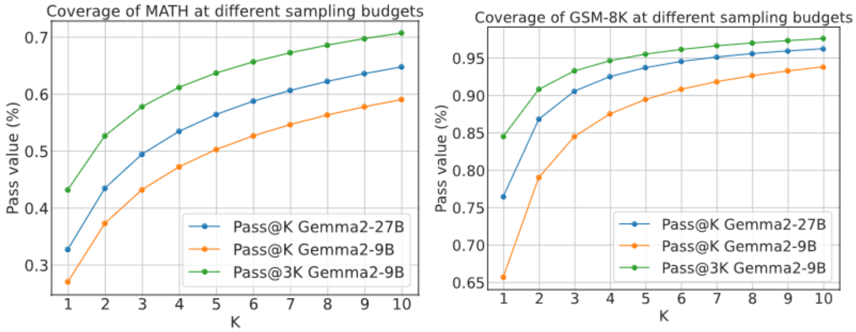
(a) Coverage on GSM-8K. (b) Diversity on GSM-8K. (c) False Positive Rate on GSM-8K.

Figure 4: Diversity, coverage, and false positive rate for the GSM-8K synthetic data [Ban+24].



(a) Finetuning Gemma-7B. (b) Finetuning Gemma2-9B. (c) Finetuning Gemma2-27B.

Figure 5: Results for models finetuned with GSM-8K synthetic data from Gemma2-9B and Gemma2-27B [Ban+24].



(a) MATH dataset.

(b) GSM-8K dataset.

Figure 6: Coverage trend for the MATH and GSM-8K synthetic datasets [Ban+24]. Pass@3K Gemma2-9B is compute-matched to Pass@K Gemma2-27B.