STANDARD DATA SCALING ANALYSES OBSCURE LOSS OF IMPORTANT INFORMATION IN MATH REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

In deep learning, increasing dataset size has been shown to improve the performance of deep neural networks. However, it is unclear if adding more data negatively impacts certain abilities learned from existing data, obscured by an overall net increase in performance. Understanding this is especially important in the current large language model era, where data scarcity has become a pressing issue. We discover that when performing fine-tuning on mathematical reasoning tasks, adding more training data causes the model to incorrectly answer a large portion of previously correctly answered test samples. This remains true even with popular test-time scaling techniques, which can iron out inconsistencies in model predictions. To better understand this phenomenon, we show that large language models fine-tuned on the same data learn very different functions across different random seeds, exhibiting extremely high predictive multiplicity. This work contains novel insights that encourage practitioners to investigate if additional data is yielding equitable benefits.

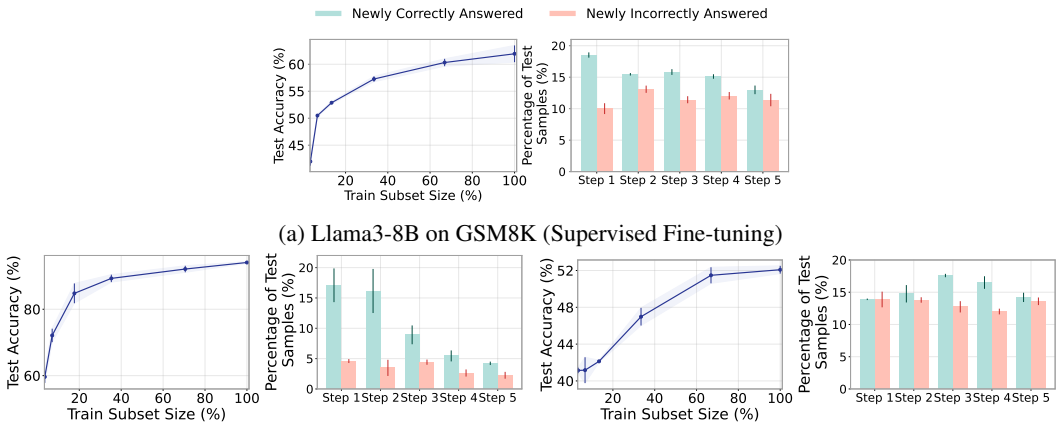
1 INTRODUCTION

It is generally understood that increasing training data can improve the performance of deep neural networks (Kaplan et al., 2020; Sorscher et al., 2022). To take advantage of this, an active effort has been applied to procure more data or generate it synthetically, specifically for settings involving large language models. However, many recent studies have shown that we are approaching the limits of human-generated text data (Muennighoff et al., 2023; Tirumala et al., 2023) and suggest that synthetic data may be a promising direction. Recent works have even attempted to study scaling laws of synthetically generated data to confirm this hypothesis, where most studies suggest that current generation techniques produce data that do not match human-generated samples. In an effort to procure more data, however, a crucial factor has been understudied: *Does adding more data* ***negatively impact*** *abilities learned from existing data, obscured by an overall net increase in performance?*

To answer this question, we begin by performing a deeper analysis of how model performance scales with an increase in training data. More specifically, rather than simply observing a net increase in test performance with increasing training data, we study how individual test samples are impacted by increasing data. Our scope is defined to cover large language model training on math reasoning tasks (Cobbe et al., 2021; Koncel-Kedziorski et al., 2016; Hendrycks et al., 2021) with supervised fine-tuning (Wei et al., 2022; Chung et al., 2022) and reinforcement learning (Shao et al., 2024). Interestingly, we observe that on addition of more training data, while test performance rises, a large portion (10-15%) of previously correctly answered test samples are now incorrectly answered with more data. This behavior is observed even with test-time scaling techniques such as majority voting, which are used to overcome inconsistencies in models' outputs produced through non-deterministic behaviors and elicit better performances (Wang et al., 2023; Brown et al., 2024; Snell et al., 2025).

With the addition of more fine-tuning data in math reasoning, large language models ***incorrectly*** answer a large portion of the ***previously correctly*** answered test samples. This is generally obscured by a ***net increase*** in test time performance.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Llama3-8B on GSM8K (Supervised Fine-tuning)

(b) Llama3-8B on MAWPS (Supervised Fine-tuning)

(c) Gemma3-4B on GSM8k (Supervised Fine-tuning)

Figure 1: With additional training data, there exist many previously correctly answered test samples that become incorrectly answered when trained using supervised fine-tuning.

We show that large language models trained on the same data but with different random seeds can learn vastly different functions, where all models have similar test performances, but the intersection of correctly answered test samples by these models is very small. Observing that large language models exhibit notably high predictive multiplicity, we link our empirical findings to prior work and show that this behavior is attributed to an inability to make full use of the training data provided. We provide an in-depth understanding of this novel failure mode of the current large language model training paradigm. We note that the objective of this work is not to provide new solutions but to offer new insights regarding the failure modes of large language models on reasoning tasks.

2 Background and Related Work

Reasoning in Deep Learning. Recent works aim to understand if deep networks are capable of drawing out underlying rules from training instances. Early works show that deep networks instead rely on approximate rules/heuristics or statistical features to solve reasoning tasks, making them unable to adapt to newer domains that rely on the same set of rules that can correctly solve the original task (Zhang et al., 2023; Liu et al., 2023; Nikankin et al., 2025). However, with the advent of more robust reasoning models, it is becoming increasingly evident that these models are capable of performing reasoning by drawing out rules and learning when to apply them to solve a problem (DeepSeek-AI et al., 2025), but still struggle to perform certain tasks.

Mathematical Reasoning. Assessing and understanding the mathematical reasoning ability of deep neural networks has become an important area of study within the field of deep learning (Zhou et al., 2024; Lee et al., 2024). (Lample & Charton, 2020) show that deep networks attain good performances in slightly complex mathematical tasks such as symbolic integration and the solving of differential equations. Following this, Cobbe et al. (2021) showed that heavily parameterized models fail when solving math word problems with simple arithmetic tasks and that attaining good performance generally requires the utilization of additional compute, especially during test time. Hendrycks et al. (2021) further reinforced this claim and showed that in word problems with more complex mathematical operations, deep networks are unable to perform. Today's reasoning models, although less parameterized than the ones discussed in (Cobbe et al., 2021), are better at solving these tasks but still struggle to solve many problems within these tasks, even with test-time scaling.

Math Reasoning Datasets. We consider 3 math reasoning datasets in this work: MAWPS (Koncel-Kedziorski et al., 2016), which comprises simple math word problems, GSM8K (Cobbe et al., 2021) which comprises harder grade school math word problems, and MATH (Hendrycks et al., 2021; Zeng et al., 2025), which comprises much harder high school math word problems. These datasets are representative of most math reasoning tasks studied in current literature.

Inclusion of Reasoning Steps in Training Data. To improve the mathematical reasoning ability of deep networks, most datasets comprise of chain-of-thought reasoning steps, where instead of standard

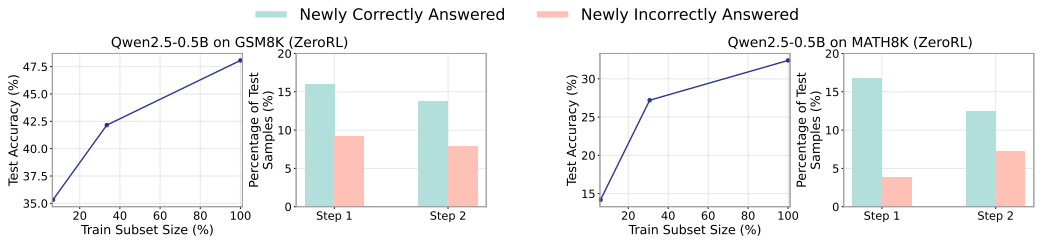


Figure 2: With additional training data, there exist many previously correctly answered test samples that become incorrectly answered when trained using reinforcement learning.

question-answer pairs, solutions often also contain intermediate reasoning steps that eventually arise at the answer. Inclusion of such intermediate reasoning steps has been shown to improve performance significantly. In addition to these, many works propose augmenting or transforming training instances to induce new ways of reasoning about mathematical word problems. Weng et al. (2023); Jiang et al. (2024) introduce backward reasoning questions for training instances, where models are taught to reason about a question better by starting from the answer and working back towards an initially provided value. Yu et al. (2024) aim to augment a training set with rephrased questions that arrive at the same answer, thereby allowing models to draw out similarities between questions that seem different on the surface.

Scaling Test-time Compute. Practitioners have shown that scaling test-time compute helps improve performance significantly on math reasoning tasks. Cobbe et al. (2021) show that by training verifiers and performing ranking of many candidate solutions, deep networks were capable of attaining much better performances on mathematical reasoning benchmarks. Wang et al. (2023) follow a simple approach of sampling many candidate solutions and simply picking the most common answer, also known as majority voting. Others use reward models to score candidate solutions and simply pick answers with the highest score (Snell et al., 2025).

Supervised Fine-tuning (SFT). Supervised fine-tuning (Wei et al., 2022; Chung et al., 2022) is the de facto training objective used to fit large language models to specific tasks because it is relatively inexpensive and does not need verification. In supervised fine-tuning, the model is trained for a few training steps to mimic the training distribution by learning to predict the next token given prior context. We define the supervised fine-tuning objective as follows:

Given a pretrained causal LM (p_θ) and a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ of inputs/prompts x_i and target outputs $y_i = (y_{i,1}, \ldots, y_{i,T_i})$, SFT minimizes the token-level negative log-likelihood (teacher forcing):

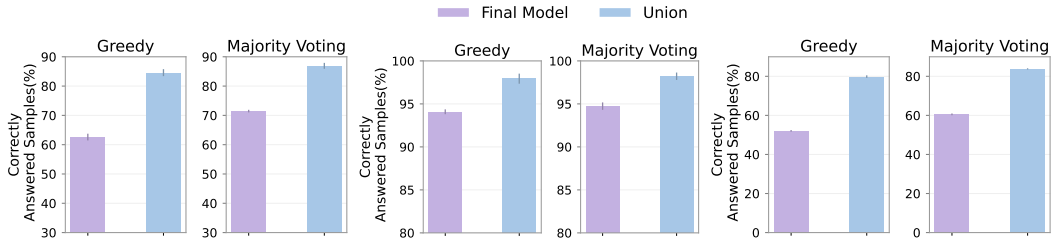
$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_i} w_{i,t} \, \log p_\theta\big(y_{i,t} \,\big|\, x_i,\ y_{i,<t}\big),$$

where $w_{i,t} \in \{0, 1\}$ masks out prompt tokens (and any tokens we choose not to supervise), and $y_{i,<t} = (y_{i,1}, \ldots, y_{i,t-1})$.

Due to its indispensability and general reliability, it is critical to study how supervised fine-tuning makes use of the data it is provided.

Reinforcement Learning (RL) with verifiable rewards. Recent work has shown that simple reinforcement learning with verifiable rewards on base language models can elicit strong reasoning behaviors (DeepSeek-AI et al., 2025; Zeng et al., 2025; Gandhi et al., 2025). Large language models are commonly trained with reinforcement learning through Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and thus, it is the technique used for RL training in this paper. More specifically, we perform ZeroRL, which is RL training on the base model without any prior supervised fine-tuning step (Zeng et al., 2025).

Parameter-Efficient Fine-Tuning. Parameter-efficient fine-tuning techniques were created to make it more efficient to fit a model to a particular task. More specifically, it freezes the weights of a model and trains a much smaller set of extra parameters to reduce memory requirements. LoRA (Hu et al., 2022), QLoRA (Dettmers et al., 2023), and LoftQ (Li et al., 2024) are the primary PEFT techniques used in practice. In this work, we make use of LoRA and LoftQ.



(a) Llama3-8B on GSM8K (b) Llama3-8B on MAWPS (c) Gemma3-4B on GSM8K

Figure 3: The union of all correctly classified test samples by models trained on different train subset sizes is significantly greater than the number of correctly classified samples by the model trained on the entire dataset.

3 Standard Scaling Law Analyses Obscure Loss of Important Information

We begin our analysis by performing supervised fine-tuning of large language models (Llama3 (Dubey et al., 2024), Gemma3 (Team et al., 2025)) on math reasoning tasks (GSM8K (Cobbe et al., 2021), MAWPS (Koncel-Kedziorski et al., 2016). To understand how additional data improves test-time performance, we fine-tune the model on increasing subsets of the dataset, where every time we increase the subset size, the new subset is always a superset of the previous subset. Note that the subsets are consistent across different random seeds. We also note that the model is fine-tuned from scratch each time we increase the subset size, instead of continually fine-tuning as we add more data. Consistent with previous analyses, as one scales the size of the train dataset, test-time performance improves, with large initial increases that plateau over time (Fig. 1 (line plots)), thereby mimicking standard scaling laws (Kaplan et al., 2020; Sorscher et al., 2022). Note that for initial experimentation, we perform greedy decoding in our evaluation.

We observe, however, that for every step at which we increase the subset-size considered, a significant number of samples that were previously correctly answered are now incorrectly answered (Fig. 1 (bar charts)). In fact, at every step in most settings, more than 10% of the test dataset that was previously correctly answered becomes incorrectly answered (**Newly Incorrectly Answered**). More interestingly, as we continue to increase the size of the dataset used to fine-tune the model, the number of samples that are newly correctly answered becomes very similar to the number of samples that are newly incorrectly answered, thereby resulting in marginal improvements in testing accuracy. We observe such patterns even when continually fine-tuning, as shown in Fig. 11. This occurs even with RL-based training, where we perform ZeroRL of Qwen2.5-0.5B (Yang et al., 2024) on GSM8K and Math8K (Hendrycks et al., 2021; Zeng et al., 2025) datasets using GRPO (Fig. 2). Note that we perform greedy decoding in all of these experiments during inference, such that there is no randomness during evaluation.

To better understand this phenomena, we compare the testing accuracy obtained by the model trained on the entire dataset (`Final Model`) against a testing metric where we assume a test sample is correctly answered if at least one of the models (trained on increasing subsets of the dataset) answers it correctly (`Union`) (see Fig. 3). We observe that the testing accuracy obtained by `Final Model` is significantly lower than that by `Union`.

Sample-level analysis. We observe that the samples that are flipped from being correctly answered to newly incorrectly answered at least once have longer generated solutions on average, having a solution of length of 109.7 tokens against the 91.54 tokens of samples that are never newly incorrectly answered. This is even true of the ground truth solution lengths - 112.32 against 92.21. This indicates that samples that are typically flipped from correctly answered to incorrectly answered require longer reasoning traces, giving more room to make more mistakes. We do not observe significant differences in output confidences across the two sets of samples - with the former having an average token margin of 0.891 and the latter having an average token margin of 0.906.

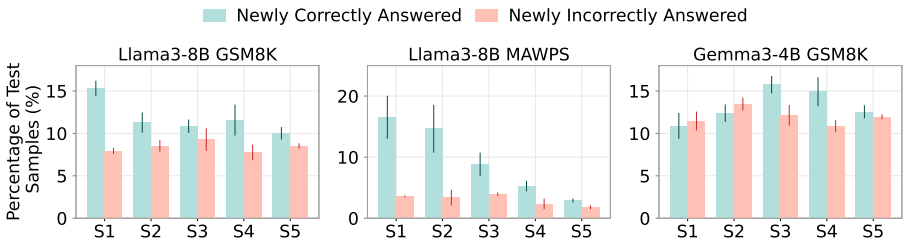


Figure 4: Even with majority voting, the addition of new training samples causes previously correctly answered samples to be incorrectly answered. `Sn` denotes step number n when increasing the train set size.

3.1 Test-time Scaling does Not Resolve the Problem

Recent work has shown that during inference, models exhibit a high degree of non-determinism, where small changes in the sampling strategy can cause big changes in model output. Thus, it is generally recommended to sample multiple outputs using temperature sampling from the same model and perform majority voting across these outputs to get the model's prediction (Wang et al., 2023). Such sampling enhances consistency in a model's prediction, overcoming minor fluctuations that can lead to incorrect answers. Such a technique falls under a broader pool of strategies known as test-time scaling techniques, used to overcome inconsistencies and even boost performance at times.

The results in Fig. 1 are presented with naive single-sample greedy decoding. To confirm that our insights are not observed simply due to these fluctuations during inference, we repeat our evaluation with majority voting (Wang et al., 2023). Consistent with standard practice (Wang et al., 2023; Brown et al., 2024), we sample 15 outputs and perform majority voting with a temperature of 0.6 and with no nucleus sampling. We observe that even with majority voting, samples continue to be incorrectly answered with the addition of more training data(Figs. 3 & 4).

We note that we only stick to majority voting for test-time scaling. Our sole objective is to overcome inconsistencies in model prediction, to understand if these inconsistencies and minor fluctuations are the root cause behind the phenomenon of **Newly Incorrectly Answered** samples in Fig. 1. We learned above, however, that this is not the case. There exist many test-time scaling techniques that can be used to enhance model performance, but Brown et al. (2024) show that these often provide competitive or worse performance than majority voting in the settings concerning math reasoning.

3.2 Experimental Details

3.2.1 Supervised Fine-Tuning

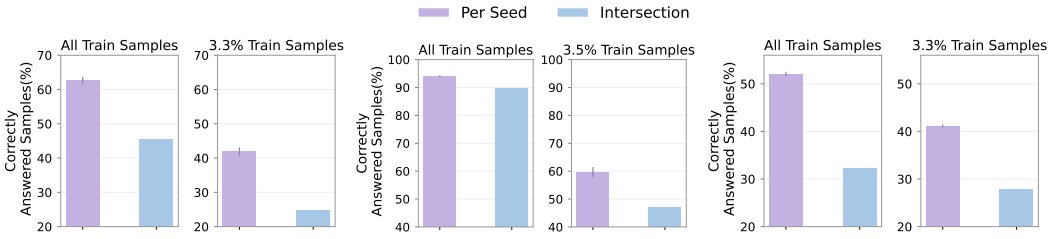
Llama3-8b GSM8K Setting. We fine-tune a Llama3-8B on the GSM8K dataset (Cobbe et al., 2021) using LoftQ (Li et al., 2024). Consistent with their implementation, we train for 3 epochs using a learning rate of 5e-4 with cosine decay and weight decay 0.1. Fine-tuning was performed on a machine with two Nvidia RTX4090s for 3 seeds.

Llama3-8b MAWPS Setting. We fine-tune a Llama3-8B on the MAWPS dataset (Koncel-Kedziorski et al., 2016) using LoftQ. We use the same hyperparameters used for the Llama3-8b-GSM8K Setting. Fine-tuning was performed on a machine with two Nvidia RTX4090s for 3 seeds.

Gemma3-4b GSM8K Setting. We fine-tune a Gemma3-4B on the GSM8K dataset using LoRA (Hu et al., 2022). We train for 5 epochs using a learning rate of 5e-4 with cosine decay and weight decay of 0.1. Fine-tuning was performed on a machine with two Nvidia RTX4090s for 3 seeds.

3.2.2 Reinforcement Learning

Qwen2.5-0.5B GSM8K Setting. We fine-tune a Qwen2.5-0.5B on the GSM8K dataset in ZeroRL fashion (Zeng et al., 2025) using GRPO (Shao et al., 2024). We train for 1 epoch using a learning rate of 5e-6 with cosine decay and weight decay of 0.1. Training was performed on a machine with one Nvidia H100 for 1 seed. We do not use any PEFT techniques.



(a) Llama3-8B on GSM8K (b) Llama3-8B on MAWPS (c) Gemma3-4B on GSM8K

Figure 6: Fixed Set Analysis - Models trained on the same dataset but across different seeds have similar testing accuracies but correctly answer very diverse test samples.

Qwen2.5-0.5B MATH8K Setting. We fine-tune a Qwen2.5-0.5B on the MATH8K dataset (Hendrycks et al., 2021; Zeng et al., 2025) in ZeroRL fashion using GRPO. We train for 1 epoch using a learning rate of 5e-6 with cosine decay and weight decay of 0.1. Training was performed on a machine with one Nvidia H100 for 1 seed. We evaluate on the MATH500 dataset (Lightman et al., 2024; Zeng et al., 2025), which is a representative subset of the MATH test set. We do not use any PEFT techniques.

4 Understanding Why Samples are Incorrectly Answered with More Data

At first glance, the results in Fig. 1 can be perceived as though additional training data is causing previously correctly answered samples to be incorrectly answered due to some seemingly conflicting information between the newly added training samples and the previously available training samples. The model is unable to disambiguate the conflict that may be present in the data. We note, however, that this is not the case. When we perform the same experiment over multiple different seeds, the newly incorrectly answered samples are very different across models. In other words, adding the *same training samples* causes different sets of test samples to be incorrectly answered.

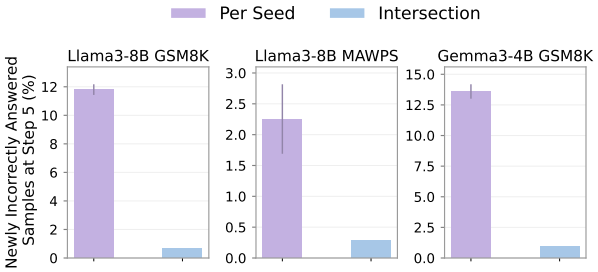


Figure 5: The intersection of the newly incorrectly answered samples at step 5 across three seeds is less than one-fifth of the average number of newly incorrectly answered samples at step 5 in all settings.

In our setting, the intersection of all the newly incorrectly answered sets by models obtained at Step 5 in Fig. 1 is less than one-fifth of all the samples that are newly incorrectly answered, as shown in Fig. 5. Due to the randomness in the newly incorrectly answered samples, it becomes evident that this occurs not because of sample-to-sample conflicts. This raises the following question: *If additional training data does not conflict with previously attained data, then why are so many previously correctly answered samples now incorrectly answered?*

4.1 Fixed Set Analysis

Since additional training samples are not causing any test samples to be newly incorrectly answered due to conflicts, we hypothesize that the model is simply learning very different functions every time it encounters the same training data but across different random seeds. To confirm this, we train the same Llama3-8B on the entire train dataset but with different random seeds. More specifically, we train these models on the same sample set of the GSM8K dataset across three random seeds. We observe that even if one were to train these models on the same fixed dataset but across *different random seeds*, they would correctly answer widely different samples within the test set, as shown

in Fig. 6. In other words, the model encodes different functions over different training runs, even if it is trained on the same data. This remains true even if the sample set used to train is only a small fraction (3.3-3.5%) of the entire train set, as shown in Fig. 6. Note that we provide the intersection of correctly classified samples across the three seeds. **Thus, while the training data can be used to correctly answer many samples within the test set, each model only correctly answers a much smaller subset of the test set. This shows that these models are unable to make complete use of their training data.**

4.2 Large Language Models Exhibit High Predictive Multiplicity on Math Reasoning Tasks

In this section, we aim to better understand why these models, trained on the same data, correctly answer a very diverse set of test samples. First, we show how this phenomenon ties into prior work and explain why training LLMs on reasoning tasks creates models that learn very different functions. Finally, we perform ablation studies, which show that simple changes to the training recipe can cause significant changes in the final learned function. We also study the role of capacity and PEFT techniques in our observations.

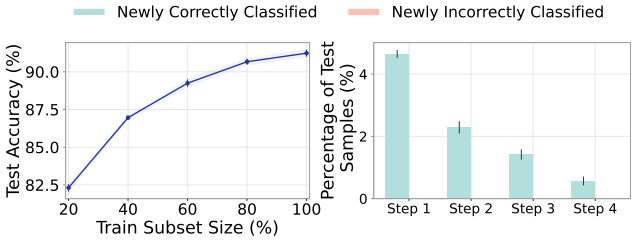


Figure 7: In simple classification tasks, scaling does not incur incorrectly classifying previously correctly classified samples. (ResNet20 on CIFAR-10)

Training on Math Reasoning Tasks Leads to Large Rashomon Sets. We observe that all models trained on the same data but with different random seeds obtain almost the same testing accuracies but differ significantly on per-sample correctness, leading to a low intersection on correctly answered samples. Such predictive multiplicity has been well documented in literature, commonly referred to as the *Rashomon effect* (Breiman, 2001; Marx et al., 2020; Semenova et al., 2023; Rudin et al., 2024). The Rashomon effect is a phenomenon where multiple models obtain similar empirical risk but differ significantly in per-sample predictions. A Rashomon set (also known as ε-level set) is the set of all models that obtain similar empirical risk constrained by some slack ε.

Definition 4.1 (Rashomon set)**.** Let $S = \{(x_i, y_i)\}_{i=1}^n$ be a dataset and $\hat{R}_S(h)$ the empirical risk. For a baseline empirical risk minimizer $h_0 \in \arg\min_{h\in\mathcal{H}} \hat{R}_S(h)$ and $\varepsilon \geq 0$, the Rashomon set is defined as

$$S_\varepsilon(h_0) \;=\; \{\, h \in \mathcal{H} \;:\; \hat{R}_S(h) \leq \hat{R}_S(h_0) + \varepsilon \,\}.$$

where $\mathcal{H}$ is the hypothesis space for that task and h is any model within the hypothesis space.

Note that in our setting, we define the Risk $\hat{R}$ as 0-1 error on the test dataset S as follows:

$$\hat{R}_S(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[\, h(x_i) \neq y_i \,].$$

Definition 4.2 (Discrepancy)**.** For every model $h \in S_\varepsilon(h_0)$, we define discrepancy as

$$\delta_\varepsilon(h) = \sum_{i=1}^{n} \mathbf{1}[\, h(x_i) \neq h_0(x_i) \,].$$

LLMs Trained on Reasoning Tasks have a Large Discrepancy. Deep neural network training takes place on a high-dimensional loss landscape, where there can be many local minima that a model may settle on that can even have the same or similar loss. It is for this reason that minor changes in the training process can cause different functions to be learned. Such behavior is well known. However, this behavior was not considered harmful as the discrepancy $\delta_\varepsilon(h)$ was often too small, such that a practitioner could be certain that these networks were reliably grasping similar functions every time they encountered the same training data. In Fig. 7, we show that if we were to conduct

similar scaling experiments with simple classification tasks, such as ResNet20 trained on CIFAR-10, one would not observe Newly Incorrectly Answered samples at all.

The Reason Behind Large Discrepancy. The core reason behind this large discrepancy is that there is a significant mismatch between the desired outcome and the loss functions used to train the model. Take for instance, supervised fine-tuning, where we attempt to minimize next token loss during training. This training paradigm is then paired with the rich and complex task of mathematical reasoning where it is difficult to find invariant reasoning rules across samples that generalize well, along with the fact that there are many strategies that can be used to correctly solve a problem. Such training does not actually minimize loss in a way that is directly meaningful to the reasoning task, that is focused on forming the right reasoning chains and pairing the correct set of operands with operators. This results in very different functions learned, all of which solve different subsets of the test data while attaining similar losses.

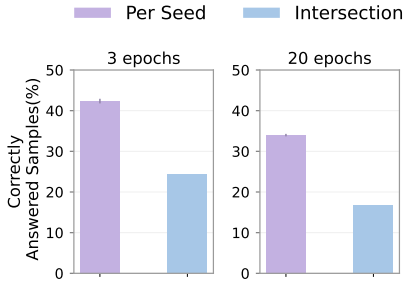


Figure 8: Training for more epochs reduces overall test accuracy but still does not reduce the discrepancy set, as is observed by a low intersection.

Finally, it is important to note that LLM fine-tuning is typically only done for a few epochs to prevent overfitting, typically ranging from 1 to 5 epochs, and thus, these models are never allowed to reach a converged state. Such practices explain why even RL-based fine-tuning techniques exhibit the issue highlighted in this paper. While this practice will evidently contribute to a large discrepancy set, we do not believe it is the primary contributor. In other words, one cannot simply fix this problem by training for longer. To show this, we maintain two settings where we train a LLama3-8B on a small subset of the GSM8K training set (128 out of 7,473 samples). In one setting, we only train for the standard 3 epochs. In the other setting, we train for 20 epochs and push the training loss to near zero. We perform this experiment for three seeds per setting and evaluate all models on the entire GSM8K test set, where we assess standard test accuracy but also the intersection of correctly answered samples per seed. Consistent with our previous claims, training for more epochs significantly hurts test accuracy by nearly ten points, as shown in Fig. 8. We also observe that the intersection of correctly answered samples across seeds per setting is significantly lower than the testing accuracy, implying that the discrepancy is still large, even on training for more epochs while hurting testing accuracy.

Collectively, this tells us that standard training techniques used to fine-tune LLMs make it such that these models are not able to make full use of the math reasoning training data to attain test accuracies that would be possible if they were able to make complete use of the training data.

4.3 Ablations

Factors Affecting Variance. Given the large discrepancy across random seeds, it must follow that even minor changes to the training recipe while keeping the dataset and all training relevant parameters the same should result in models that differ significantly. We confirm this experimentally. Through extensive ablations in the Llama-GSM8K setting, we discover that there are two factors due to which we observe the learning of very different functions:

- **Sample Order.** It is well documented within the deep learning literature that the order in which samples are processed during training plays a very important role in the final function map that is eventually learned. We observe that changing this order across runs but keeping other factors constant encourages the learning of very different functions.
- **LoRA Dropout.** In our LoftQ (Li et al., 2024) setup,

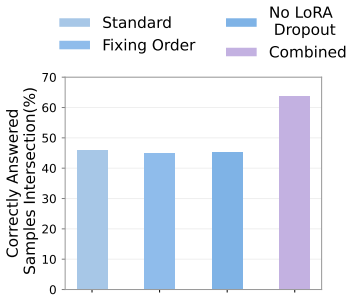


Figure 9: Discarding or fixing factors affecting randomness in the training process makes the model learn the same function across seeds. It is evident by the high intersection of correctly answered samples of `Combined`. `Combined` represents fixing the sample order & discarding the LoRA dropout. Intersection is taken across 3 seeds.

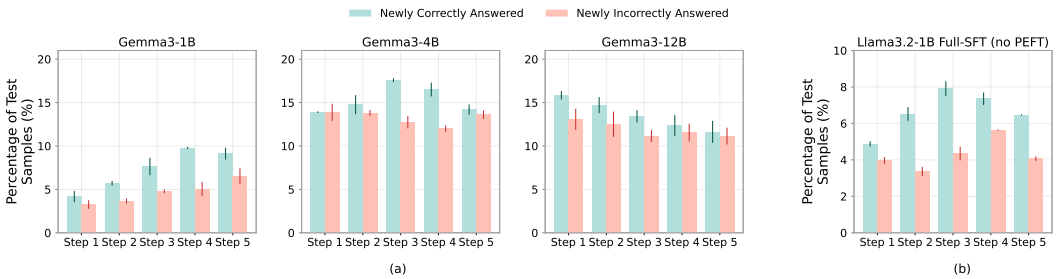


Figure 10: (a) Samples continue to be newly incorrectly answered with the addition of more training data, despite an increase in capacity. (b) Samples are still newly incorrectly answered with the addition of more training data, even without the use of PEFT techniques (Llama3.2-1B on GSM8K).

we maintain a LoRA dropout ratio of 0.1, whose application varies from one run to another based on the random seed. Such variance causes training to be driven in different directions, ultimately resulting in very different functions being learned.

We observe that on fixing the order in which the model observes samples during training across three seeds, while also removing the LoRA dropout, the model learns the same function, resulting in a large intersection of correctly classified samples, as shown in Fig. 9 (the size of intersection set is equal to the size of correctly answered sample set of any of the three models). **These experiments show how small changes in the training recipe (sample order and LoRA dropout application) can cause models to learn widely different functions.**

Role of Capacity. We test if model capacity plays a role in scaling and if models of larger capacity tend to be better at making better use of the training data. To answer this, we train 3 models of different capacities (1B, 4B, and 12B) but belonging to the same model family (Gemma3). We observe that models of all capacities face the same issue, where adding more training data causes test samples to be newly incorrectly answered, as shown in Fig. 10 (a). This happens despite these models having different generalization capabilities, attaining final test accuracies of 18.57%, 53.98%, and 72.93%, respectively.

Role of PEFT techniques. We also test if our observations are consistent, even if one were not to make use of PEFT techniques and instead perform full supervised fine-tuning. We observe that even when performing full supervised fine-tuning, adding more training data causes samples to be newly incorrectly answered, as shown in Fig. 10 (b).

5 Conclusion

In this work, we answer an important question: *Does adding more data negatively impact abilities learned from existing data, obscured by an overall net increase in performance?* First, we observe that model performance scales poorly with increasing dataset sizes, where adding more data causes smaller improvements over time due to the incorrect prediction of previously correctly answered samples. We then show that this happens not because of sample conflicts but because these models exhibit high predictive multiplicity on math reasoning tasks, where the same models trained on the same data but across different seeds learn very different functions. These models attain similar testing accuracies, but they correctly answer very different sets of test samples. Thus, while the training data can be used to correctly answer many samples within the test set, each model only correctly answers a small subset of the test set. This shows that these models are unable to make complete use of their training data. Finally, we tie this observation to prior work and explain why Large Language Model training on math reasoning tasks is bound to have high predictive multiplicity.

Limitations. We note that we do not perform any analysis on pre-training data and focus only on fine-tuning data. Studying how well these models make use of pre-training data is outside the scope of this work and computationally infeasible for most practitioners.

AI USE STATEMENT

AI was used for writing code and polishing the plots within this work. AI was also used for generating irrelevant context, as described in the appendix.

REPRODUCIBILITY STATEMENT

We provide training and evaluation details within this work. Upon acceptance, all code and evaluation sets used within the paper will be open-sourced.

REFERENCES

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–215, 2001. ISSN 08834237, 21688745.

Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787, 2024. doi: 10.48550/ARXIV.2407.21787. URL https://doi.org/10.48550/arXiv.2407.21787.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022. doi: 10.48550/ARXIV.2210.11416. URL https://doi.org/10.48550/arXiv.2210.11416.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen

Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *CoRR*, abs/2503.01307, 2025. doi: 10.48550/ARXIV.2503.01307. URL https://doi.org/10.48550/arXiv.2503.01307.

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.

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*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James T. Kwok. Forward-backward reasoning in large language models for mathematical verification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 6647–6661. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.397. URL https://doi.org/10.18653/v1/2024.findings-acl.397.

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. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 1152–1157. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/N16-1136. URL https://doi.org/10.18653/v1/n16-1136.

Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1eZYeHFDS.

Nayoung Lee, Kartik Sreenivasan, Jason D. Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dsUB4bst9S.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LzPWWPAdY4.

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*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=De4FYqjFueZ.

Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6765–6774. PMLR, 13–18 Jul 2020.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=O9YTt26r2P.

Cynthia Rudin, Chudi Zhong, Lesia Semenova, Margo I. Seltzer, Ronald Parr, Jiachang Liu, Srikar Katta, Jon Donnelly, Harry Chen, and Zachery Boner. Position: Amazing things come from having many good models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=oFDFGd9Age.

Lesia Semenova, Harry Chen, Ronald Parr, and Cynthia Rudin. A path to simpler models starts with noise. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Uzi22WryyX.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. URL https://doi.org/10.48550/arXiv.2402.03300.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=UmvSlP-PyV.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. D4: Improving LLM pretraining via document de-duplication and diversification. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=CG0L2PFrb1.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=s4xIeYimGQ.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin

Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. On the paradox of learning to reason from data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3365–3373. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/375.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua M. Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AssIuHnmHX.

A APPENDIX

A.1 CONTINUAL FINE-TUNING

We observe that even with continual fine-tuning, samples are still newly incorrectly answered with the addition of more data, as shown in Fig. 11.

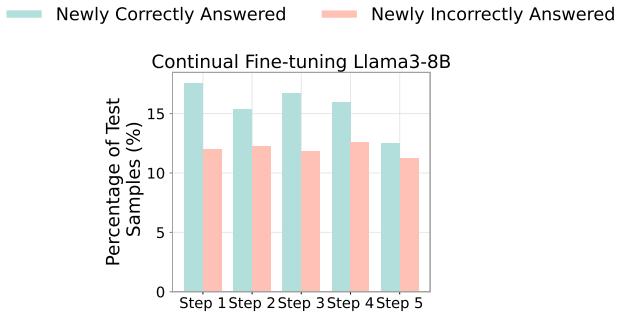


Figure 11: Samples are still newly incorrectly answered, even with continual fine-tuning (Llama3-8B on GSM8K).