Quality or Quantity? Comparing Domain-Adaptive Pre-training Approaches for Language Models with Mathematical Understanding

Stanford CS224N Custom Project

Christine Ye
Department of Physics
Stanford University
`cye@stanford.edu`

Alexandre Acra
Department of Mathematics
Stanford University
`acra@stanford.edu`

Abstract

For even the most sophisticated models, mathematical reasoning remains incredibly challenging. Our project investigates whether pre-training on math-related texts can improve performance of broadly capable models. We test two approaches: single-epoch pre-training on extremely large domain-specific corpora such as `MathPile` and `DeepMind Mathematics`, and quality-based curriculum-based pre-training on mathematics textbooks (`textbooks` from `MathPile`) using a novel perplexity ratio score. We find that single-epoch pre-training on massive corpora actually degrades MMLU performance across the board compared to model baselines. However, multi-epoch pre-training on high-quality mathematics textbooks significantly improves performance on MMLU mathematics tasks compared to both baseline models and models pre-trained on a non-math corpus. Curriculum-based pre-training, particularly training on documents ordered by quality or even on just the top 25% highest-quality documents, further improves performance, even when models are trained on smaller datasets and for less compute time. Our results suggest that *text quality is more important than text volume or compute time* for mathematics-specific pre-training, and that using curriculum pre-training on the highest-quality documents can be highly effective.

1 Team Information

- TA mentor: Yuan Gao
- Team Contributions: Christine worked on pre-training benchmark evaluation, and writing. Alexandre worked on qualitative and quantitative comparisons of model performance, embedding space analysis, and writing.

2 Introduction

As language models rapidly improve their capabilities for problem-solving, abilities of mathematical reasoning are of keen interest due to their broad applicability both for mathematics in itself, and applications to any other quantitative problem-solving. One main factor driving improvements in LLMs' performance on mathematical reasoning is the data they are pretrained on, and the way in which this pretraining is structured. Yuan et al. (2023) find that enhanced arithmetic and problem-solving skills come from including code and LaTeX in pretraining corpora, motivating both math-specific pretraining and a question of quality and makeup of data included in such corpora.

While many current approaches to pretraining rely on improvements via more data and more parameters, we are interested in investigating the question of quality versus quantity in pretraining data; more specifically, can approaches of curricular pretraining emphasizing quality and order of data used improve model accuracy on mathematics reasoning as hypothesized? Moreover, can domain-specific pretraining in general improve LLMs' performance for mathematical reasoning tasks?

To answer these questions, we aimed to employ mathematically-oriented pretraining to improve capabilities for mathematical reasoning, with approaches ranging from general pretraining on corpora centered around mathematics, to curricular pretraining with the idea of simulating progressive skill math development.

To provide us with insight on *quality*, our first experiment pretrained `GPT-Neo` on full math-relevant corpora including MathPile and DeepMind's mathematics dataset, and evaluating these pretrained models on tasks in the MMLU dataset compared to `GPT-Neo` model baselines of various sizes.

To provide us with insight on *quality*, our second experiment was pretraining GPT-Neo on various subsets of these math-relevant corpora based on curricular pretraining selections, using our novel constructed perplexity ratio metric, which optimizes for comparatively more difficult data for GPT-Neo, through sorting of the order of data in pretraining based on this score, and selection only of data with high score values.

In comparing the results of these two experiments, we found notably higher performance for the quality end of our question, with curricular pretraining on perplexity ratio metric-oriented section of corpora yielding much high model accuracy. Further qualitative analysis confirms this improvement anecdotally, with solution generation demonstrating improved capacity for understanding of question content, and embedding analysis suggesting that improvements came from beyond developments in arithmetic and numerical understanding.

3 Related Work

3.1 Domain-adaptive pre-training

Models pretrained on large and diverse corpora can be fine-tuned with much greater efficiency than models trained from scratch Huang et al. (2020). However, the corpora used for pre-training do not cover the full range of human knowledge domains. As a result, significant work has considered *domain-adaptive pre-training* (DAPT), which allows models to continue training on an additional corpus of in-domain text. For example, Gururangan et al. (2020a) perform domain-adaptive pre-training on RoBERTa across biomedical publications, CS publications, news, and reviews (10 - 50GB corpora), showing that DAPT consistently improves RoBERTa's performance on in-domain tasks. In practice, a number of models trained with DAPT have been highly successful in in-domain applications, such as Llemma for mathematics and LegalXLMs for law Azerbayev et al. (2024) Niklaus et al. (2024). Their findings imply that in-domain pretraining for mathematics is a viable method to improve mathematical understanding of language models, prompting our work.

However, domain-adaptive pre-training can have negative consequences for performance on out-of-domain tasks, in what is known as *catastrophic forgetting* (see Kirkpatrick et al. (2016), Riemer et al. (2018)). In catastrophic forgetting, domain-specific pre-training, especially in multiple stages or highly specific domains, causes performance deterioration on general-purpose benchmarks like MMLU. This suggests that the content of domain-specific corpora could significantly impact DAPT results; even pre-training on higher-order mathematics, for example, could theoretically deteriorate model understanding of elementary arithmetic.

3.2 Curriculum pretraining

Substantial previous research has also considered whether *curriculum pre-training* – training models on documents in a certain order, usually based on difficulty – can improve pre-training efficacy for a given corpus. For example, Shi et al. (2024) find that pre-training using sequences of related documents within a context window substantially improves in-context reasoning, especially across long ranges. Wang et al. (2020) use a difficulty-based curriculum for speech translation, in which models are sequentially trained on more advanced courses, and also find substantial performance improvements. However, the BabyLM Challenge, which trained models on small, fixed data budgets, generally found that curriculum-based approaches showed only modest gains and were less effective than, say, student-teacher models Warstadt et al. (2023). Thus it is still unclear which tasks do or do not benefit from curricula; we aim to study if mathematical reasoning, in particular, can benefit.

3.3 Data quality

While today's highest-performing language models are pre-trained on trillions of tokens, there has been substantial debate on what kinds of corpora are most effective. On one hand, Warstadt et al. (2023) and Zhang et al. (2020) find that models learn high-fidelity representations, and can even outperform models trained on massive corpora, with just 10M to 100M tokens. Furthermore, Micheli et al. (2020) find that past a critical training data volume, more pre-training data does not substantially improve performance on downstream French language QA tasks. On coding and problem-solving tasks, `Microsoft Phi-1` and `Phi-2`, trained only on $\sim$ 7B tokens of high-quality textbooks, outperform models trained for hundreds of billions of tokens, including `GPT-3.5`. These findings suggest that high-quality, smaller corpora may be more effective for pre-training.

On the other hand, Komatsuzaki (2019) and Xue et al. (2023) argue that single-epoch pre-training on a larger corpus is more effective than longer pre-training on smaller corpora – essentially, that quantity is king. In addition, research into scaling laws for large language model find that loss scales with parameter count, compute, and dataset size, suggesting larger datasets lead to better performance Kaplan et al. (2020). Our work tests both sides of this debate, comparing pre-training on large corpora with small, high-quality curriculum-based corpora.

4 Approach

4.1 Model

For our experiments, we use EleutherAI's GPT Neo models with 125M parameters Black et al. (2021), which is based on the architecture of Generative Pre-trained Transformer 2 (GPT-2) decoder model Radford et al. (2019). We compare the performance of the 125M-parameter model to the larger GPT-Neo-1.3B, with 1.3 billion parameters. Across these parameter sizes, GPT-Neo typically ranges from 16 to 32 layers, with a default hidden size of 2048, and a default number of 16 attention heads per layer, with GeLU activation functions and layer normalization for improved performance and stability. GPT-Neo is trained on The Pile Gao et al. (2020), which consists of 22 diverse subsets ranging from PubMed to OpenWebText2. In particular, the Pile includes arXiv papers and the DeepMind Mathematics dataset, which Gao et al. (2020) find significantly improves model bits-per-byte on math-focused texts. We utilize the `HuggingFace transformers` library for our GPT-Neo implementation and training loop Wolf et al. (2020).

To evaluate the performance of domain-adaptive pretraining for mathematical texts, we utilize two baseline models: default GPT-Neo and GPT-Neo pretrained on a novel, non-mathematics corpus for a standardized batch size and number of training steps. Comparing to these baselines will allow us to understand whether domain-adaptive pretraining can be effective in mathematics.

4.2 Domain-adaptive pretraining on massive corpora

Some previous work has suggested that dataset volume is most important for pre-training, and that pre-training for few epochs – even just 1 – is the optimal approach for large corpora. Therefore the first approach we consider is domain-adaptive pretraining on massive corpora, using just 1 training epoch. We select multiple large ($\sim$ billions of tokens) mathematics-focused corpora to pre-train on, and utilize a different domain-specific corpus as a control; we compare all outputs to the baseline `125M` and `1.3B` models. We pretrain on domain-specific corpora using an autoregressive objective, in which the model trains to causally predict the next token based on previous context.

4.3 Curriculum pre-training approach

We also consider a curriculum-based training approach. For a model with parameters θ, we define the perplexity P of a document X with T tokens as

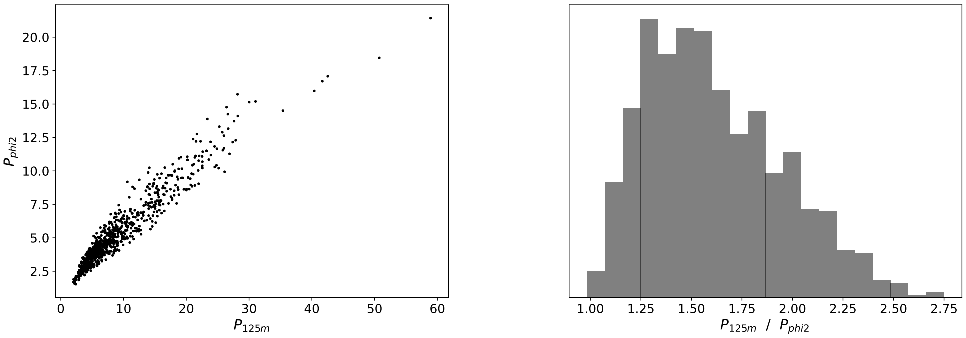
$$P(X) = \exp\left(-\frac{1}{T}\sum_{i}^{T}\log p_\theta(x_i|x_{<i})\right)$$

In general, documents with higher perplexity are "more difficult" for the model than documents with lower perplexity. Extremely difficult documents, such as graduate-level mathematics textbooks in `MathPile`, have high perplexity, and would not be particularly useful to pretrain the model on. On the other hand, documents that are extremely simple may contain lower-quality information, and may not improve the model's mathematical understanding. Instead, we aim to preferentially pretrain on documents with high *learning potential*, which are currently out-of-domain for the model but of reasonable difficulty such that pretraining can actually be effective.

To devise a curriculum of high-quality documents with high learning potential, we devise and implement a novel *perplexity ratio metric*. In this scheme, perplexities of documents in the pretraining corpus are computed both with `GPT-Neo-125M` and some larger, highly mathematically capable model, which we choose to be `Microsoft Phi-2`. We compute the *perplexity ratio (PR)* between the model pair, thus assigning higher scores to documents which are difficult for `Neo-125M` but less difficult for `Phi-2`. As illustrated in Figure 1, we find that document perplexity in `Neo-125M` roughly correlates with perplexity under `Phi-2`, as expected. We also find that scores are with a slight right skew, suggesting there is a long tail of documents with high learning potential.

We then preferentially pretrain models based on perplexity ratio scores using two approaches. Our first approach, *sorting*, orders documents by perplexity ratio such that the model trains on high-PR documents first. Our second approach, *top-k*, segments the top 25% and 50% of documents, and *only* trains the model on this subset. For an equal amount of compute time, we compare performance of models trained on *top-k* documents with models trained on the full corpora.

Due to compute constraints, we do not consider multi-stage curricula in which corpora change between epochs, and we perform all curriculum experiments on the `textbooks` subset of `MathPile`.



(a) `GPT-Neo-125M` vs. `Phi-2`

(b) Perplexity ratio

Figure 1: Perplexity scores (P) for `MathPile` documents

5 Experiments

5.1 Data

For pretraining on massive corpora, we use two large math-specific corpora and one non-math domain-specific corpus as a control. The first math-specific corpus is`MathPile`, a billion-token-scale corpus (∼30GB) designed specifically for math (Wang et al. (2023)). `MathPile` is deduplicated, filtered, and includes diverse high-quality mathematics texts from arXiv, Wikipedia, ProofWiki, StackExchange, and high-quality textbooks.To contrast with the highly abstract content in `MathPile`, which includes many graduate-level textbooks and research-level papers, we use the Deepmind Mathematics Dataset, a synthetic ∼ billion-token high-school mathematics question-answer dataset Saxton et al. (2019). Finally, as a control corpus, we use the Pile of Law, which has minimal overlap with both the original Pile and any mathematics-based corpus, and can serve as an out-of-domain control (Henderson* et al. (2022)). We load and stream datasets using the `huggingface datasets` library.

5.2 Evaluation method

For standardized evaluation, we use the Massive Multitask Language Understanding (MMLU) benchmark Hendrycks et al. (2021), focusing on performance in Formal Logic ($N = 145$), Elementary Mathematics ($N = 424$), and High School Mathematics ($N = 304$). We evaluate using EleutherAI's open-source Language Model Evaluation Harness implementation (Gao et al. (2023)), which uses zero-shot prompting and evaluates the model on the full multiple-choice answer with the highest decoder score.

5.3 Experimental details

To pretrain on massive domain-specific corpora, we train `GPT-Neo-125M` for 1 epoch on each corpus: `MathPile`, `DeepMind Mathematics`, and `Pile of Law`. For curriculum pre-training, we train on a subset of MathPile (`textbooks`) and an equal-sized subset of Pile of Law (`us-bills`). Since our curricular approach changes corpus sizes, we standardize our experiments' compute time to 12.5k steps; to account for overfitting on smaller datasets, we also report MMLU evaluation results at 5k steps. For all experiments, following the experimental setup in Gururangan et al. (2020b), using mixed-precision training, the Adam optimizer with a weight decay of 0.01, learning rate 1e-5, and linear learning rate decay. Given computation contraints we use a batch size of 4, and train with ZeRO Stage-2 (using `deepspeed` Rasley et al. (2020)).

5.4 Results

As a baseline, regardless of parameter count, GPT-Neo models generally perform only slightly better than random (25%) on mathematics-relevant baselines, and sometimes worse than random. We report results from training on `MathPile`, `Pile of Law`, and `DeepMind Mathematics` in Table 1. Evaluation results from curricular pretraining on `textbooks` from `MathPile` are shown in Table 2. Results are displayed both from the full training, as well as at an earlier checkpoint, to address possible overfitting, since the top-25 and top-50 subsets are significantly smaller than the full `textbooks`. We focus on the following MMLU tasks: EM: elementary mathematics; HSM: high school mathematics; STEM: all science, technology, engineering, and mathematics tasks; CM: college mathematics; AA: abstract algebra. We also include HUM (all humanities tasks) as a baseline, which should not be significantly affected by pretraining.

We find that single-epoch pre-training on large corpora is largely ineffective in improving MMLU performance. Pre-training on mathematics corpora degrades performance compared to the baseline model across all domains, from elementary mathematics to humanities. Interestingly, models pretrained on `Pile of Law`, a completely different domain, actually out-perform all other models on college mathematics and abstract algebra. Although models may still see some qualitative improvement in subject matter understanding, this does not translate to MMLU scores.

However, we find that multi-epoch pre-training on smaller corpora, using a curriculum approach, does result in performance improvements. While pre-training on our control corpus, `us_bills`, mostly degrades performance across MMLU tasks and particularly mathematics tasks, pre-training on `textbooks` significantly improves performance on elementary and college mathematics. In fact, performance on high school mathematics and college mathematics exceeds the `GPT-Neo-1.3B` baseline. We find that sorting `textbooks` by perplexity ratio improves performance over the unsorted corpus across all tasks except abstract algebra. Finally, our top-25 and top-50 pre-training approaches see substantial improvements in certain subject areas. For example, the model trained on the top 25% of documents for 5k steps beats the `1.3b` baseline in both high school mathematics and abstract algebra. Interestingly, pre-training on textbooks also slightly improves performance on MMLU-HUM, our control evaluation task; however, the changes in mathematics task performance are much more substantial.

Table 1: Base and Simple Pretraining Results (Better than 125M Base Bolded, Best Blue)

Model	**MMLU-EM**	**MMLU-HSM**	**MMLU-STEM**	**MMLU-HUM**	**MMLU-CM**	**MMLU-AA**
125M Base	0.225 ± 0.022	0.219 ± 0.025	0.215 ± 0.007	0.244 ± 0.006	0.240 ± 0.043	0.190 ± 0.039
Mathpile	0.217 ± 0.021	0.211 ± 0.025	0.213 ± 0.007	0.241 ± 0.006	0.220 ± 0.042	$\mathbf{0.210 \pm 0.041}$
Mathpile 10k	0.222 ± 0.021	0.207 ± 0.025	0.214 ± 0.007	0.240 ± 0.006	0.240 ± 0.043	$\mathbf{0.210 \pm 0.041}$
Deepmind	0.209 ± 0.021	0.211 ± 0.025	0.212 ± 0.007	0.242 ± 0.006	0.210 ± 0.041	$\mathbf{0.210 \pm 0.041}$
Deepmind 10k	0.212 ± 0.021	0.211 ± 0.025	0.213 ± 0.007	0.242 ± 0.006	0.210 ± 0.041	$\mathbf{0.210 \pm 0.041}$
Lawpile	0.201 ± 0.021	0.215 ± 0.025	0.214 ± 0.007	0.241 ± 0.006	$\mathbf{0.260 \pm 0.044}$	$\mathbf{0.230 \pm 0.042}$
Lawpile 10k	0.217 ± 0.021	0.211 ± 0.025	0.214 ± 0.007	0.243 ± 0.006	0.220 ± 0.042	$\mathbf{0.210 \pm 0.041}$
1.3B Base	0.283 ± 0.023	0.259 ± 0.027	0.280 ± 0.008	0.250 ± 0.006	0.290 ± 0.046	0.220 ± 0.042

Table 2: Curricular Pretraining Results (Better than 125M Base Bolded, Best Blue)

Model	MMLU-EM	MMLU-HSM	MMLU-STEM
125M Base	0.225 ± 0.022	0.219 ± 0.025	0.215 ± 0.007
US Bills	0.212 ± 0.021	0.211 ± 0.025	$\mathbf{0.217 \pm 0.007}$
Textbooks	$\mathbf{0.259 \pm 0.023}$	$\mathbf{0.226 \pm 0.025}$	$\mathbf{0.237 \pm 0.008}$
Textbooks 5k	$\mathbf{0.243 \pm 0.022}$	$\mathbf{0.237 \pm 0.026}$	$\mathbf{0.229 \pm 0.007}$
Textbooks Top25	0.217 ± 0.021	$\mathbf{0.241 \pm 0.026}$	$\mathbf{0.251 \pm 0.008}$
Textbooks Top25 5k	$\mathbf{0.241 \pm 0.022}$	$\mathbf{0.263 \pm 0.027}$	$\mathbf{0.248 \pm 0.008}$
Textbooks Top50	$\mathbf{0.254 \pm 0.022}$	$\mathbf{0.256 \pm 0.027}$	$\mathbf{0.233 \pm 0.008}$
Textbooks Top50 5k	$\mathbf{0.241 \pm 0.022}$	$\mathbf{0.230 \pm 0.026}$	$\mathbf{0.236 \pm 0.008}$
Textbooks Sorted	$\mathbf{0.265 \pm 0.023}$	$\mathbf{0.230 \pm 0.026}$	$\mathbf{0.239 \pm 0.008}$
1.3B Base	0.283 ± 0.023	0.259 ± 0.027	0.280 ± 0.008

Model	MMLU-HUM	MMLU-CM	MMLU-AA
125M Base	0.244 ± 0.006	0.240 ± 0.043	0.190 ± 0.039
US Bills	0.242 ± 0.006	0.220 ± 0.042	$\mathbf{0.220 \pm 0.042}$
Textbooks	$\mathbf{0.251 \pm 0.006}$	$\mathbf{0.320 \pm 0.047}$	0.190 ± 0.039
Textbooks 5k	$\mathbf{0.245 \pm 0.006}$	$\mathbf{0.250 \pm 0.044}$	0.160 ± 0.037
Textbooks Top25	$\mathbf{0.258 \pm 0.006}$	$\mathbf{0.300 \pm 0.046}$	$\mathbf{0.200 \pm 0.040}$
Textbooks Top25 5k	$\mathbf{0.263 \pm 0.006}$	$\mathbf{0.250 \pm 0.044}$	$\mathbf{0.250 \pm 0.044}$
Textbooks Top50	$\mathbf{0.253 \pm 0.006}$	0.210 ± 0.041	$\mathbf{0.270 \pm 0.045}$
Textbooks Top50 5k	$\mathbf{0.247 \pm 0.006}$	$\mathbf{0.250 \pm 0.044}$	$\mathbf{0.230 \pm 0.042}$
Textbooks Sorted	$\mathbf{0.254 \pm 0.006}$	$\mathbf{0.320 \pm 0.047}$	0.170 ± 0.038
1.3B Base	0.250 ± 0.006	0.290 ± 0.046	0.220 ± 0.042

6 Analysis

6.1 Qualitative examples

To qualitatively analyze the impact of our mathematics-specific pre-training approaches, we use the following prompt and compile responses from each model variant.

Prompt

The fundamental theorem of calculus states

Model Responses

`125M`: $\int_{\mathbb{R}^n} \frac{d^n x}{(x-y)^n} = \int_{\mathbb{R}^n}$

`top-25`: If X is a real number
Theorem 1.1.1. If X is a real number, then
X is a real number
Proof. Let a be a real number. Then

`MathPile`: that the set of all functions f on a set X is the set of all functions f on X such that $f(x) = x$ for all $x \in X$.

`LawPile`: 1. The function
(a) The function
(b) The function
(c) The function

`1.3b`: The derivative of a function is the rate of change of the function.

This prompt lies in the domain of high school and college mathematics, so we can compare the responses from different pretraining strategies which improved vs. did not improve performance in these domains. We can first notice that none of these responses are correct; however, they are incorrect in quite different ways.

The baseline 125M model gives an incomplete equation without a theorem statement—the given response does contain notation of calculus, but with no coherent meaing or relation to the FTC.

The top-25 curricular model, which had previous improvements over the 125m baseline in college and high school math accuracy, answers the prompt in a more targeted way (by attempting to state a mathematical theorem in a way that resembles standard theorem and proof openings at this level of mathematics), but its answer does not contain any of the content of the FTC, matching the notable improvement but still relatively low performance seen in the table above.

The model pretrained on MathPile presents a grammatically correct mathematical sentence, but with no mathematical meaning or relation to the FTC—training on the full corpus could give a strong sense of mathematical language, but not true understanding of meaning, matching the low performance above.

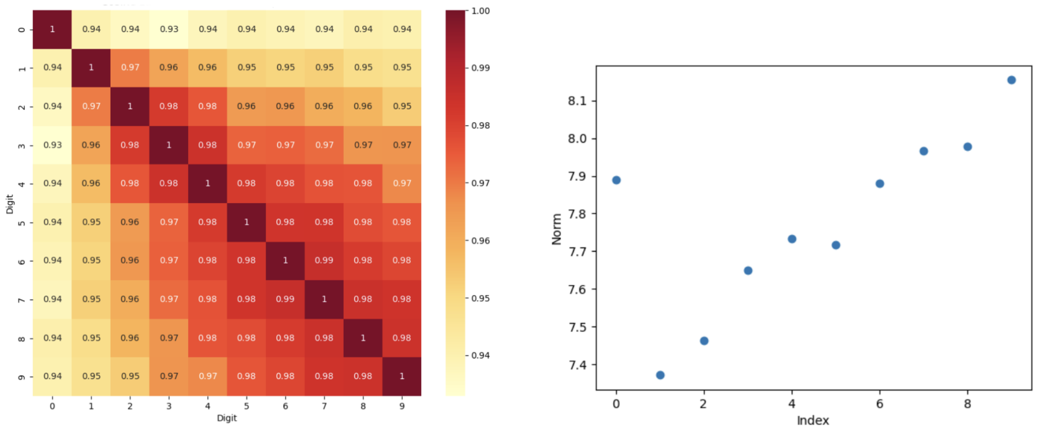
The model pretrained on LawPile (negative control) presents an answer with almost no mathematical meaning and little grammatical cohesion, matching the overall negative performance seen generally.

Finally, the 1.3B-parameter GPT-Neo baseline gives a grammatically coherent and correct sentence related to the contents of calculus, although it is not the correct FTC—this matches previous higher accuracy than most other models, but still low accuracy.

6.2 Embedding space structure

Previous work suggests that transformer-based language models perform better on mathematical tasks, particularly arithmetic, when certain embedding space schemes are enforced for math-related tokens (McLeish et al. (2024)). In particular, Golkar et al. (2023) created an embedding scheme for numerical data such that all numbers were represented by a single token, with the embedding vector scaled by number value, such that the embedding space is continuous for numerical inputs. Motivated by this work, we seek to understand whether such a continuous vector space structure for number tokens already exists in pre-trained models, even if not explicitly enforced in the architecture. We investigate of the vector embeddings of digits 0-9 in both baseline and pretrained `125M` models, for potential consequences in mathematical performance, by looking at pair-wise cosine similarities and vector norms, to understand whether the vector space of numerical tokens is indeed continuous.

In Figure 2a, we see that all digits have relatively high pairwise cosine similarity in their embeddings. However, we see a higher cosine similarity between digits with higher values. This can likely be attributed to higher digits primarily appearing in mathematical contexts, versus lower digits having more multi-context usage. For instance, more phrases exist in the English language involving "1" and "2" with unique meanings than for higher digits, which are used in almost exactly equivalent contexts. Figure 2b, which plots digit vector norms, also suggests a linear relationship between the magnitude of embeddings and digit values, therefore suggesting largely parallel and numerically scaled vectors. While the slope of this linear relationship is not 1 (meaning that for instance, summing the vectors for two numbers will not come close to the vector for their sum), the linear trend suggests digits are embedded continuously based on value.



(a) Cosine Similarity Heatmap for Digit Embeddings

(b) Scatter Plot of Digit Embedding Norm versus Numerical Value

Figure 2: Analysis of Similarity of Digit Embeddings in Direction and Magnitude

We also find that cosine similarity and norm values were virtually identical for the 125m baseline of GPT-Neo and all pretrained variants (hence only one of each plot being shown), meaning that the embeddings for digits largely did not change with pretraining. Thus we claim that changes to *digit* embeddings is not a key contributing factor to performance improvements in higher-level mathematics. Moreover, this either suggests that GPT-Neo already embeds digit values quite effectively, or if there is another optimum embedding structure to be found, it is not achieved by pretraining on large math corpora, either normally or curricularly.

7 Conclusion

Overall, we find that while single-epoch pre-training is largely ineffective in improving performance on math-related tasks, curriculum-based pre-training across multiple epochs does improve model performance on mathematical benchmarks. This suggests that training on a subset of the highest-quality data can be more effective than the full corpus. We also find that embeddings for numerical tokens, in both baseline and pre-trained models, exhibit a continuous vector space structure.

We conjecture that our models trained on large, math-specific corpora may be suffering from catastrophic forgetting, leading to benchmark performance deterioration. Although our pre-training experiments on large corpora see training loss saturation within 1 epoch, suggesting the models are well-fit, post-training evaluation on MMLU exhibits worse results than our baseline model, even on relevant topics such as college mathematics and abstract algebra. This true across the board, regardless of dataset. However, our qualitative analysis does suggest that regardless of MMLU results, pre-training still moves the model distribution closer to the corpus.

Multi-epoch pre-training on mathematics textbooks without a curriculum approach already produces significantly better results than baseline or pre-training on large corpora, despite `textbooks` being a subset of `MathPile`. This aligns with previous work ex. Gunasekar et al. (2023) and suggests that high-quality textbooks are an effective corpus for mathematics pre-training. Using curriculum approaches based on the perplexity ratio score, we find that for a given corpus and amount of compute, simply sorting the documents can improve model performance. We also find that training on a smaller, high-quality subset of the original `textbooks` corpus provides comparable, if not better, performance on many tasks. For example, training on just the top 25% highest-perplexity ratio documents actually exceeds training on the full corpus for abstract algebra, elementary mathematics, and high school mathematics. This suggests that the majority of pre-training results can actually be achieved with smaller, high-quality datasets and less compute time. This also suggests that preferentially pre-training on high *learning potential* documents might reduce performance degradation or catastrophic forgetting. In general, our results suggest that *quality* is more important than *quantity* for mathematics-specific DAPT at a given level of compute, or even with less compute time.

We note that our work has several limitations. First, MMLU mathematics tasks are not a perfect benchmark for language model mathematical ability, as they do not account for factors such as qualitative model reasoning. In addition, since `GPT-Neo`'s performance is already only slightly better than random to begin with, it is difficult to draw strong conclusions about model abilities. Furthermore, due to compute constraints, our work draws conclusions only on the efficacy of pre-training on large, domain-specific corpora for a single epoch; it is possible that multiple epochs of training, or other datasets, could produce improvements above baseline. Finally, our work only considers the effect of curricular pretraining using a relatively small dataset, and does not fully explore the impact of different datasets, quality metrics, or training hyperparameters. Avenues for future research could include varying these factors, particularly the quality metric, or using a staged curricular approach, where based on this metric, the model would be trained on different subsets in different epochs.

8 Ethics Statement

One potential ethical challenge of our project comes from training on biased data, especially in mathematics. One well known phenomenon is that names of those from underrepresented backgrounds are often featured less in math word problems, creating issues of representation for students of those backgrounds. Day-to-day ramifications could include students feeling less represented in mathematics courses when they don't see their cultures and names represented in real-world contexts of math problems, or even more dangerously, biased narratives being integrated into math word problems, like stereotypical names being attributed to certain actions in the problems, which can hurt students' feelings of inclusion and spread harmful narratives in young minds. For instance, if problems and solutions generated were to associate women's names with stereotypically feminine activities, this could hurt young girls' feelings of inclusion in mathematics, lowering participation and potential in the field later on. In order to mitigate this, we can select and compensate in our training data for equitable representation of different backgrounds and scenarios, and ensure that the model is not trained on data that might contain harmful stereotypes in its language.

One negative societal risk of a very competent math problem-solving model is that it could incentivize overreliance on it, rather than developing your own creative solutions as a student, which could harm overall educational potential. More specifically, if students no longer learn how to solve math problems on their own, or only solve problems in one certain way influenced by bias in the model, potential for mathematical problem-solving in the next generation could be hurt, which would have wide ramifications for societal potential for progress in mathematics and other quantitative disciplines. In order to mitigate this, selecting for generation of incomplete information (ex: extra steps in providing solutions, including developments of hints and general problem-solving strategies) could temper immediate reliance on full solutions for users. In order to hypothetically implement this from a technical standpoint, one could use prompt engineering strategies to only have the model engage with the user in certain ways. For instance, a model accessible to students could be pre-prompted with the fact that it is working with a student, and should only give hints and partial solutions to help the student learn. Currently, with available models, if at the beginning of a conversation, instruction guidelines are given for how to behave in a given conversation, they can be respected quite rigorously—this could be automatically enforced in models given to students so that they only engage with them with language appropriate and conducive to learning.

References

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Siavash Golkar, Mariel Pettee, Michael Eickenberg, Alberto Bietti, Miles Cranmer, Geraud Krawezik, Francois Lanusse, Michael McCabe, Ruben Ohana, Liam Parker, Bruno Régaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. 2023. xval: A continuous number encoding for large language models.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020a. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020b. Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

Peter Henderson*, Mark S. Krass*, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2020. Clinicalbert: Modeling clinical notes and predicting hospital readmission.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

Aran Komatsuzaki. 2019. One epoch is all you need.

Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, and Tom Goldstein. 2024. Transformers can do arithmetic with the right embeddings.

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel E. Ho. 2024. Multilegalpile: A 689gb multilingual legal corpus.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *ArXiv*, abs/1810.11910.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvasy, Rich James, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2024. In-context pretraining: Language modeling beyond document boundaries.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020. Curriculum pre-training for end-to-end speech translation.

Zengzhi Wang, Rui Xia, and Pengfei Liu. 2023. Generative ai for math: Part i – mathpile: A billion-token-scale pretraining corpus for math.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling llm under token-crisis.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks?

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R. Bowman. 2020. When do you need billions of words of pretraining data?