ITC 1/54

Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 345-358
DOI 10.5755/j01.itc.54.1.39713

Synthetic Data Enhances Mathematical Reasoning of Language Models Based on Artificial Intelligence

Received 2024/12/06 | Accepted after revision 2025/02/18

HOW TO CITE: Han, Z., Jiang, W. (2025). Synthetic Data Enhances Mathematical Reasoning of Language Models Based on Artificial Intelligence. *Information Technology and Control,* 54(1), 345-358. https://doi.org/10.5755/j01.itc.54.1.39713

Synthetic Data Enhances Mathematical Reasoning of Language Models Based on Artificial Intelligence

Zeyu Han

Data Science and Analytics, Georgetown University; 3700 O St., N. W., Washington, D. C. 20057, United States; e-mail: zh272@georgetown.edu

Weiwei Jiang

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications; Beijing 100876, China; e-mail: jww@bupt.edu.cn

Zeyu Han and Weiwei Jiang are co-first authors of the article

Corresponding authors: zh272@georgetown.edu

Current large language models (LLMs) training involves extensive training data and computing resources to handle multiple natural language processing (NLP) tasks. This paper endeavors to assist individuals to compose feasible mathematical question-answering (QA) language models in specific fields. We leveraged Gretel.ai, a feasible data generation platform, to generate high-quality mathematical QA data covering several areas, including definitions, theorems, and calculations related to linear algebra and abstract algebra. After fine- tuning through Open-AI infrastructure, GPT-3 performed significant improvements on accuracy, achieving a roughly 18.2% increase in abstract algebra benchmark, approximately 1.6x improvement on linear algebra theorems benchmark, and approximately 24.0% increase on linear algebra calculations benchmark. And small language models (SLMs) such as LLama-2-7B/13B and Mistral-7B have outstanding around 2x accuracy advancements in linear algebra calculations. This study demonstrates the potential for individuals to develop customized SLMs for specialized mathematical domains using synthetic data generation and fine-tuning techniques.

KEYWORDS: AI generated data; artificial intelligence; text classification; data collection cost; mathematical question-answering; downstream task training.

1. Introduction

1.1. Background

In recent years, there has been a significant improvement in NLP and LLMs techniques to increase the comprehensive ability and generalization of models. From word embedding models [37, 43] to Transformer based encoder and decoder autoregressive models [13, 46, 4, 7, 52, 51], the flourishing progress of LLMs depends on appearance of Transformer structure [53], innovation of effective finetuning algorithms and techniques [26, 10, 19], and the gradually increasing diversity and scale of training data.

In order to improve the ability of LLMs, [24] indicated that the model's performance could be enhanced by increasing its parameters to enlarge the model size to improve performance according to abundant database. However, the cost of computational resources, primarily GPUs, and data collection increases proportionally with the size of the model. Fine-tuning a sparse Mixtral model with 2M queries may require a NVIDIA H100 GPU with cost of $3460 [58]. And pre-training a LLM is substantially more expensive, sometimes reaching millions of dollars, due to requirements of GPU clusters, massive dataset, and electric consumption. Taking GPT-3 175B [4] as an example, it is trained on V100 GPU high-bandwidth clusters with mixed datasets composed of CommonCrawl [46] and WebText [45] totaling nearly 430 billion tokens and its training expenses exceed $4.6 million [27].

Meanwhile, data quality has become an area of concern. In the case of unsupervised pre-training, the quality of training data involved in few-shot learning process would greatly affect the performance of LLMs, thus influencing the generalization and adaptability of models to different downstream tasks [4]. Similarly, training LLMs with adequate AI generated NLP feedback data and efficient parametric fine-tuning technique LoRA [19] could effectively improve the performance of QA task in low-data scenarios [30].

Therefore, this paper initially proposes to address the downstream task by utilizing AI generated high-quality data to verify the effectiveness of our method in QA of mathematical definitions, theorems and calculations. On the one hand, our method could effectively reduce the costs associated with data collection, data cleaning, and computing resources. The synthetic data could be generated effortlessly without quantity limitation and tailored to meet diverse requirements for different applications in various domains. On the other hand, individuals could train small mathematical language models to fulfill personal demands.

1.2. Objective

Since data plays a crucial role in the fine-tuning process of downstream tasks for LLMs, the performance of models typically shows a monotonic increasing trend with the alignment degree between pre-training data and downstream task fine-tuning data [20]. In order to effectively align the downstream task data with the large amount of pre-trained data, the followings should be noted: (i) Include the relevant areas of specific targets [57]; (ii) Ensure the diversity and accuracy of data, in other words, data quality [29]. Our method could adequately explore the generalization of LLMs to ensure the performance of SLMs for specific task. Compared to LLMs, SLMs could achieve even or better performance with less computational resources, time, and size of dataset. For example, there are some highly effec- tive BERT-based SLMs:DistilBERT [47], ALBERT [28], TinyBERT [23], and MiniLM [55]. SLMs improve their performance by learning the self-attention mechanism of LLMs during the training process, forming a relationship similar to that of a teacher and students [55, 31]. Fine-tuning LLMs with a fewer well-filtered dataset, i.e., instruction fine-tuning data [29], is a practical approach that can enable the model to achieve SoTA performance on various tasks [5].

Notably, [25] conducted a study combining a simple prompt template with few-shot learning, which allowed models to gradually analyze answers and inference logic to effectively improve the models' problem-solving ability and accuracy of outputs. In addition, [54] proposed that using prompt-based data augmentation could ensure the quality of synthetic data which would increase the natural language understanding of models.

Prior research has indicated that fine-tuning mathematical tasks using AI generated instructions for existing datasets would accelerate the reasoning process and enhance model's performance. Our contributions could be summarized as follows:

- Integrate the instructions directly into data generation process which provides models with elaborate procedures of problem-solving, and facilitate customized datasets tailored to individual requirements.
- Reduce the costs of fine-tuning models by simplifying data collection and lowering the requirements for computational resources.
- Improve the accuracy of models in mathematical QA after fine-tuning and explore the influence of model choices to their performance.

2. Related Works

In recent years, general multimodal LLMs have an impressive development in many areas of NLP such as BERT series [13], T-5 [46], Megatron-Turing NLG [49], LLaMA series [51], ChatGPT series [39], PaLM [8], and these models have outstanding performance on many NLP tasks as shown in Figure 1. Meanwhile, these models also have corre- sponding SLMs such as GPT-4o mini [40], LLama-2-7B/13B [52], and TinyLlama [60]. However, these models may be unsatisfactory in some mathematical QA tasks or details. For example, ChatGPT 4o may could not correctly recognize which number is larger as shown in Figure 2.

With the rapid advancement of LLMs, mathematical problem-solving capability has emerged to be one of critical standards to evaluate the effectiveness and efficiency of LLMs. Based on well-curated pre-trained LLMs, researchers have developed multiple effective techniques to finetune models specifically for mathematically downstream tasks or building SLMs.

Figure 1

Recent years representative LLMs which achieve SoTA performance during their periods. BERT [13] initially used bidirectional Transformer architectures. T-5 [46] proposed to covert NLP tasks into a text-to-text form. GPT-3 [4] performed excellent effectiveness of few-shot and zero-shot learners and made a wide influence. PaLM [8] strongly validated the Scaling Laws in LLMs pre-training. GPT-4 [39] and LLama3 [15] are the most popular and well-known LLMs present days due to their high performance

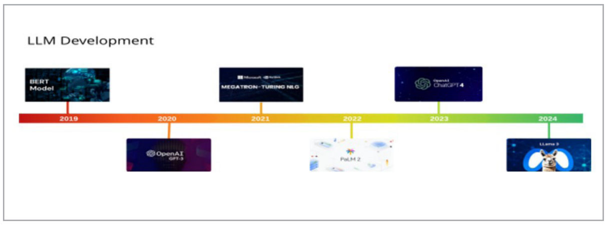
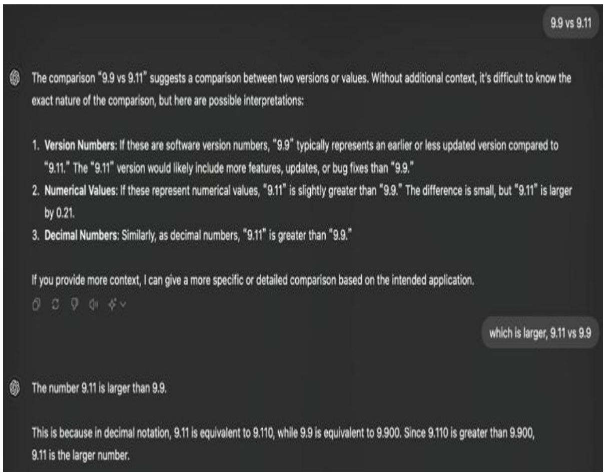


Figure 2

ChatGPT 4o sometimes may have confusion to figure out which number is larger, 9.11 or 9.9



9.9 vs 9.11

The comparison "9.9 vs 9.11" suggests a comparison between two versions or values. Without additional context, it's difficult to know the exact nature of the comparison, but here are possible interpretations:

1. **Version Numbers:** If these are software version numbers, "9.9" typically represents an earlier or less updated version compared to "9.11." The "9.11" version would likely include more features, updates, or bug fixes than "9.9."
2. **Numerical Values:** If these represent numerical values, "9.11" is slightly greater than "9.9." The difference is small, but "9.11" is larger by 0.21.
3. **Decimal Numbers:** Similarly, as decimal numbers, "9.11" is greater than "9.9."

If you provide more context, I can give a more specific or detailed comparison based on the intended application.

which is larger, 9.11 vs 9.9

The number 9.11 is larger than 9.9.

This is because in decimal notation, 9.11 is equivalent to 9.110, while 9.9 is equivalent to 9.900. Since 9.110 is greater than 9.900, 9.11 is the larger number.

AlpaGasus: Developed by [5], the AlpaGasus model represents a feasible technique that utilizing powerful LLMs to mitigate the performance reduction of Alpaca [50] caused by the misleading and detrimental IFT data. In addition, AlpaGasus achieves a remarkable cost saving which reaches $4.78 lowest for a 7B model. It emphasizes the significance of data quality for model performance.

MAmmoTH: As an instruction tuning based math model, MAmmoTH [21] primarily enhanced the general mathematical reasoning ability according to train the model on a dataset called MathInstruct that covers multiple mathematical areas and corresponding hybrid rationales. The model's performance on general math benchmarks [18, 9, 33] has a significantly improvement compared to other open source models such as WizardMath [34].

MathBERT: Unlike other models, MathBERT [42] focused on the structures of formulas and their corresponding contexts to strengthen the semantic understanding of mathematical formulas of the model during pre-training process. According to pre-training model on data including formula with context, MathBERT has demonstrated high relevance score on NCTIR-12 [59] benchmark and remarkable precision and recall on TopicMath-100K [42] benchmark.

It performed outstanding results on mathematical information retrieval, formula topic classification and formula headline generation downstream tasks.

o1-mini: On September 14th, 2024, OpenAI released the o1-mini model [41] which made a progressive advancement in cost-efficient reasoning capabilities in mathematics. Notably, o1-mini has outperformed both GPT-4o and GPT-4o-mini on the AIME benchmark, while also offering a more economical inference cost than o1 and o1-preview. Furthermore, o1-mini is 3 to 5 times faster than o-1 preview with correct answers compared to GPT-4o. However, the cost of o1-mini API would be $1000 which is expensive for individuals.

Our paper leverages the convenience and effectiveness of mathematical text generation in LLMs and cheapness of cloud computing to finetune task specific model with limited conditions for individuals. From an expenditure perspective, our method skips the instruction filtering step and straightforwardly generates high quality data compared to AlpaGasus [5] which avoids additional time consumption and charges. From an academic perspective, our method concentrates on the particular mathematical task which may be more optimal for individuals to develop a model to meet specific requirements in contrast to MAmmoTH [21] and MathBERT [42].

3. Data Description

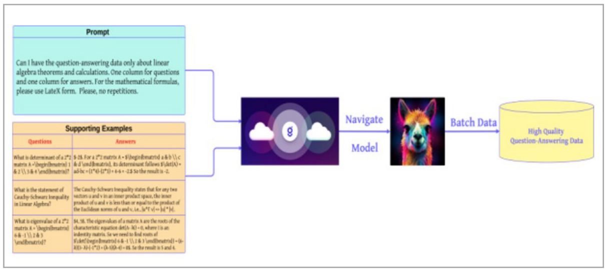
3.1. Fine-Tuning Data Generation

The data used to fine-tune the models is composed of three curated datasets with theorems and calculation of mathematics: one primarily focuses on linear algebra theorem problems (5000 rows), another on computational problems of linear algebra (3000 rows), and the third containing 3000 abstract algebra problems.

For the data generation process, as shown in Figure 3, we initially designed elaborate prompts and illustrative examples covering theorems and calculations pertinent to the specific mathematical field. This step provided language models in Gretel.ai with supplementary contexts to accomplish in-context learning. Then, the cloud platform generated 100 rows of tabular data through tunable parameters T = 1.0(temperature controlling the randomness of generation), K = 40(number of highest probability tokens considered for generation), and P = 1.0 (cumulative probability threshold for token selection) to maximize the variability of generated sample data. Subsequently, the platform leveraged existing prompt and augmented samples to construct a navigator model capable of selecting appropriate fine-tuned sub-models and generating data in batches as required as shown in Diagram 1. The linear algebra data was generated from Gretel-LLAMA-3.1-8B [36] and abstract algebra data was generated from Gretel GPT-3.5 Turbo [38]. In addition, we have standardized the mathematical formulas into LaTeX format to guarantee consistency.

Figure 3

Data generation platform Gretel.ai. We provide prompt and sample data for Gretel.ai cloud to create navigator model. According to navigator, the platform chooses the Gretel-Llama-3.1-8B-Instruct model to batch synthetic linear algebra data



Prompt

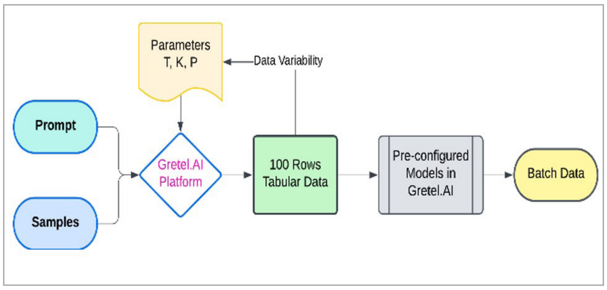
Can I have the question-answering data only about linear algebra theorems and calculations. One column for questions and one column for answers. For the mathematical formulas, please use LateX form. Please, no repetitions.

Supporting Examples

Questions	Answers
What is determinant of a 2*2 matrix A=\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}?	-2. For a 2*2 matrix A = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}, its determinant follows $\det(A) = ad-bc = (1*4)-(2*3) = 4-6 = -2.$ So the result is -2.
What is the statement of Cauchy-Schwarz Inequality in Linear Algebra?	The Cauchy-Schwarz Inequality states that for any two vectors u and v in an inner product space, the inner product of u and v is less than or equal to the product of the Euclidean norms of u and v, i.e., |u^T v| <= |u| * |v|.
What is eigenvalue of a 2*2 matrix A = \begin{bmatrix} 6 & -1 \\ 2 & 3 \end{bmatrix}?	$4, 5$. The eigenvalues of a matrix A are the roots of the characteristic equation det(A- λI) = 0, where I is an indentity matrix. So we need to find roots of $\det(\begin{bmatrix} 6 & -1 \\ 2 & 3 \end{bmatrix}) = (6-λ)(3- λ)-(-1*2) = (λ-5)(λ-4) = 0$. So the result is 5 and 4.

Diagram 1

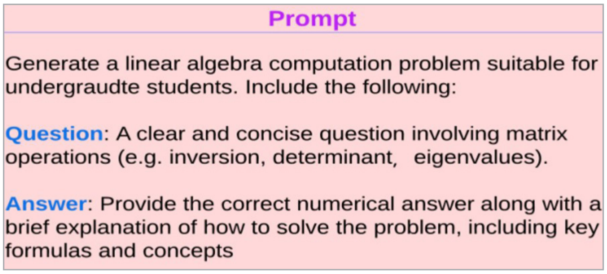
Data Generation process in Gretel.AI Platform



Nevertheless, we observed that the linear algebra dataset contains few computational problems and corresponding solutions. Although language models possess zero-shot learning capabilities [4], the lack

Figure 4

Prompt to synthesize Linear Algebra Computation QA Data



Prompt

Generate a linear algebra computation problem suitable for undergraudte students. Include the following:

Question: A clear and concise question involving matrix operations (e.g. inversion, determinant, eigenvalues).

Answer: Provide the correct numerical answer along with a brief explanation of how to solve the problem, including key formulas and concepts

of computation section would reduce the models' performance significantly. Therefore, we also used Gretel-LLAMA-3.1-8B with parameters T = 0.9, K = 35, and P = 0.8 to synthesize linear algebra calculation dataset including reasoning process containing necessary concepts and formulas according to effective prompt design as shown in Figure 4, which could be considered as data augmentation [14].

In contrast to prior research [5, 21, 25], our data generation method provides individuals with a feasible approach to obtain cost-effective high quality data, as shown in Figure 5, for fine-tuning customized models. The total time to generate the data was approximately three hours without any expenses since Gretel.ai provides all users free 1.5 million characters usage per month.

3.2. Benchmarks

In order to examine the feasibility of our fine-tuning method, we chose widely used mathematical benchmarks and took samples from them to evaluate the performance of fine-tuned models' accuracy on these benchmarks. The specific datasets we used are listed in Table 1.

Table 1

Overview of datasets and benchmarks used in the experiments

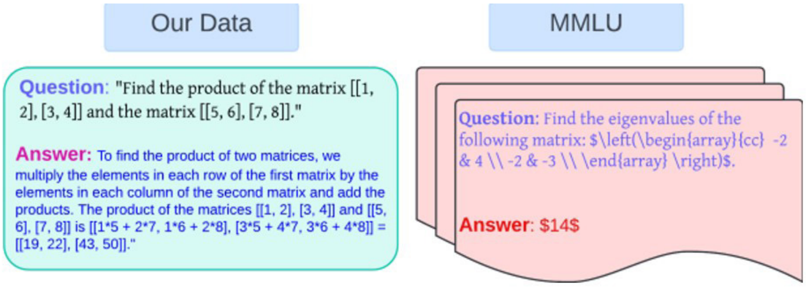
Datasets	Source	Size	Usage
Linear Algebra	Gretel LLAMA-3.1-8B	5.0k	Fine-tuning
Abstract Algebra	Gretel GPT-3.5-Turbo	3.0k	Fine-tuning
Linear Algebra Calculation	Gretel LLAMA-3.1-8B	1.0k	Fine-tuning
Theorem QA	[6]	52	Evaluation
MATH	[18]	2.0k	Evaluation
Linear Algebra QA	[32]	223	Evaluation
Partial MMLU	[17]	101	Evaluation

TheoremQA [6] is designed for evaluating the models' mathematical reasoning ability to apply theorems into specific question to deduce the correct answer. Since it lacks a dedicated linear algebra section, we utilized human evaluation to filter the satisfactory linear algebra data from algebra portion as test set.

MATH [18] is a widely used benchmark for evaluating the mathematical reasoning abilities of LLMs. It contains various areas including precalculus, algebra, geometry, and number theory, among others, as test datasets. However, the original MATH dataset does not include linear algebra QA data. In order to address

Figure 5

Data difference between two datasets. In our dataset, we included the process of solving problems, which is similar to chain-of-thought [56] to get outputs compared to MMLU



Question: "Find the product of the matrix [[1, 2], [3, 4]] and the matrix [[5, 6], [7, 8]]."

Answer: To find the product of two matrices, we multiply the elements in each row of the first matrix by the elements in each column of the second matrix and add the products. The product of the matrices [[1, 2], [3, 4]] and [[5, 6], [7, 8]] is [[1*5 + 2*7, 1*6 + 2*8], [3*5 + 4*7, 3*6 + 4*8]] = [[19, 22], [43, 50]]."

Question: Find the eigenvalues of the following matrix: $\left(\begin{array}{cc} -2 & 4 \\ -2 & -3 \\ \end{array} \right)$.

Answer: 14

this drawback and evaluate linear algebra ability of fine-tuned models, we randomly selected 1000 eigenvalue problems and determinant problems equally from the linear algebra portion of AMPS pretraining dataset where you can find it here as a dedicated test set.

Linear Algebra QA [32] dataset categorizes the difficulty of problems into five levels and provides direct answers accompanied with comprehensive explanations. Although this dataset could be suitable for pretraining or fine-tuning, its limited size of 223 rows indeed constrains the effectiveness of potential purposes due to insufficient diversity and scale.

MMLU [17] is a comprehensive benchmark covering 57 subjects across STEM to evaluate models' performance under zero-shot or few-shot settings. In the mathematics section, a subsection dedicated to abstract algebra contains multiple versions of QA data encompassing a range of topics such as group theory and ring theory.

4. Experiments

Our experiments primarily aim to achieve efficient fine-tuning of mathematical QA ability of language models while minimizing associated costs. In Section 3.1, we leveraged the Gretel.ai platform to generate high-quality synthetic datasets for linear algebra and abstract algebra without expenses and prepared them for subsequent fine-tuning procedures. In Section 3.2, we extracted the necessary data from well-established benchmarks and standardized their formats to facilitate validation.

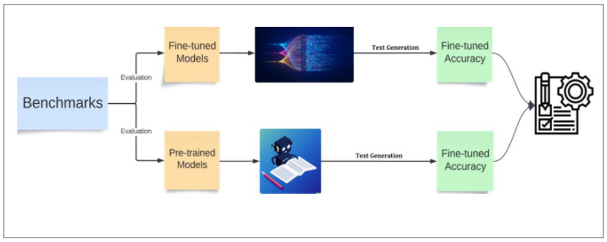
4.1. Mechanism Workflow

Initially, we deployed the pre-trained models on Google Colab utilizing an A100 GPU with 40GB of RAM to evaluate their performance on predefined benchmarks in Table 1. Subsequently, we fine-tuned these models using AI-generated mathematical datasets and reevaluated their performance to observe improvements as Figure 6 shown. We focused on two key metrics to assess the fine-tuned models:

Accuracy: The primary metric for mathematical QA tasks involving calculations and proofs. While some linear algebra and abstract algebra problems necessitate theoretical proofs, evaluating the reasonability of answers usually requires assessing the accuracy of generated answers and logical steps of the proof.

Figure 6

Workflows of our experiment



Cost-Effectiveness: To enable individuals to train personalized mathematical SLMs tailored to specific requirements as discussed in Section 1, the cost of computational resources of fine-tuning models and accessing synthesized high-quality data would be a crucial metric to justify the feasibility.

4.2. Base Models

We fine-tuned a diverse set of base language models: open-sourced small language models like LLama-2-7B/13B and Mistral due to efficiency of deployment and free of charge; and close-sourced models such as GPT-3.5-Turbo since OpenAI has provided available fine-tuning pipelines and affordable pricing.

LLAMA-2-7B/13B [51] are open-sourced auto-regressive models developed by Meta with 2 trillion pretraining to- kens, 4092 context lengths, and over 100K fine-tuning data.

Mistral-7B-v0.1 [22] is an open-sourced model developed by Mistral AI with the usage of Grouped-Query Attention [1], Sliding-Window Attention [16], and Byte-fallback BPE tokenizer [2] techniques to enhance the efficiency and performance of the model on many NLP tasks.

Bloom-7B1 [3] is a multilingual SLM developed by BigScience which is a decoder-only model modified from Megatron-LM GPT2 [48] and was trained using 8-bit optimizers [11] and ALiBI positional encodings [44].

GPT-3.5-Turbo [38] is a LLM developed by OpenAI, representing an evolution of the GPT-3 series, in other words, an enhancement of GPT-3 with advanced performance. It covers many NLP tasks including mathematical reasoning and question-answering.

4.3. Baseline Evaluation

Initially, we evaluated the base models' performance on four benchmark datasets using accuracy as the primary metric. Furthermore, we employed the GPT-4 model as a classifier to assess the alignment between the benchmarks answers and the answers generated by models to quantify the accuracy. Given our focus on the linear algebra capabilities of SLMs, we selected two benchmark datasets for our baseline assessment: Linear Algebra QA and MATH Linear Algebra.

Table 2
Accuracy of Language Models on Algebra Benchmarks

Benchmark	Model	Accuracy
MMLU Abstract Algebra	GPT-3.5-Turbo (LLM)	22.00%
		9.62%
Linear Algebra Theorem QA Linear Algebra QA (SLM) LLama-2-13B (SLM) Mistral-7B-v0.1 (SLM)	GPT-3.5-Turbo (LLM) GPT-3.5-Turbo (LLM) 5.83%	9.62% 31.84% LLama-2-7B 8.07% 14.80%
Bloom 7B1 (SLM) MATH Linear Algebra LLama-2-13B (SLM) Mistral-7B-v0.1 (SLM)	GPT-3.5-Turbo (LLM) 0.30%	0.90% 8.60% LLama-2-7B (SLM) 1.05% 1.95%
Bloom 7B1 (SLM)		0.00%

According to Table 2, we observed that the SLMs exhibited limitations in linear algebra calculations compared to GPT- 3.5-Turbo. This performance disparity might be attributed to the inherent constraints of SLMs in handling complex mathematical reasoning tasks. Furthermore, while model performance generally improves with increasing parameter size [24], our observations suggest that it is not the sole determining factor since the performance of Mistral-7B-v0.1 on both benchmarks exceeded LLaMa-2-13B.

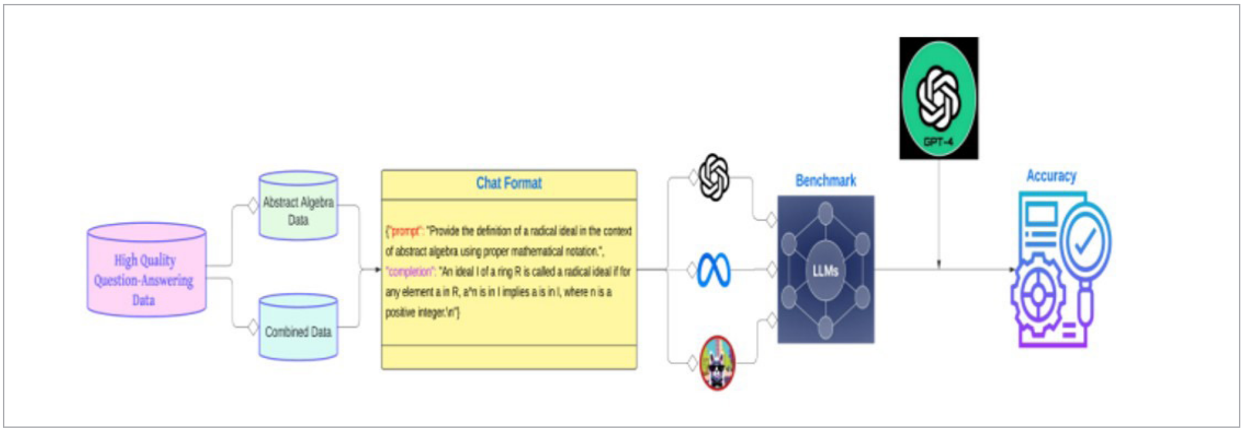
4.4. Finetuning Settings

Followed by instruction of Figure 7, we employed Huggingface AutoTrain tool to fine-tune SLMs on NVidia 1xL40S 8 vCPUs and 62GB of memory. By the way, AutoTrain has a user-friendly interface and cost-effectiveness which makes it accessible for people without coding experience.

According to GPT-3.5-Turbo requirements of training data format, we converted our CSV data into JSONL format to accommodate GPT chat-model fine-tuning requirements. Subsequently, we utilized OpenAI's API to access its infrastructure to fine-tune models with our synthetic datasets according to instructions of OpenAI Docs. And a well-structured CSV file with a single text column containing questions and corresponding answers would be sufficient for optimal fine-tuning in Autotrain. The following hyperparameters used for fine-tuning were employed:

- GPT-3.5-Turbo: Epochs = 3, Batch size = 6, and Learning rate multiplier = 2.

Figure 7
After obtaining the fine-tuning data, we separated them into two subsets: Abstract Algebra and Combined Dataset. Then we used different datasets to fine-tune models and took accuracy as our metric for evaluation according to GPT-4 model



```
{"prompt": "Provide the definition of a radical ideal in the context of abstract algebra using proper mathematical notation.", "completion": "An ideal I of a ring R is called a radical ideal if for any element a in R, a^n is in I implies a is in I, where n is a positive integer.\n"}
```

- LLama-2-7b/13b, Bloom 7B1: Default settings of Autotrain. Chat template = none, Mixed precision = fp16, Optimizer = adamw torch, LORA = True, Scheduler = Linear, Batch size = 2, Block size = 1024, Epoches = 3, Gradient accumulation = 4, Learning rate = 0.00003, Model max length = 2048.
- Mistral-7B-v0.1: We adjusted the hyperparameters from the previous configuration, increasing the batch size to 3 and the number of epochs to 4 for better accommodation of model.

The Autotrain and OpenAI's API platforms provided us convenient and efficient fine-tuning approaches for users to train language models.

4.5 Results

According to Figure 8, we observed that the fine-tuned model not only provided correct answers but also offered explanations, aligning with the Chain-of-Thought reasoning approach [56]. We fine-tuned the GPT-3.5-Turbo model on two distinct datasets: one consisting exclusively of abstract algebra data, and the other comprising a combination of abstract algebra, linear algebra, and linear algebra calculation data. Both fine-tuned models have performed remarkable progresses on benchmarks. However, as shown in Figure 9, we unexpectedly observed that the model exclusively fine- tuned on abstract algebra data

Figure 9

Performance of GPT-3.5-Turbo and its fine-tuned models across various datasets and benchmarks

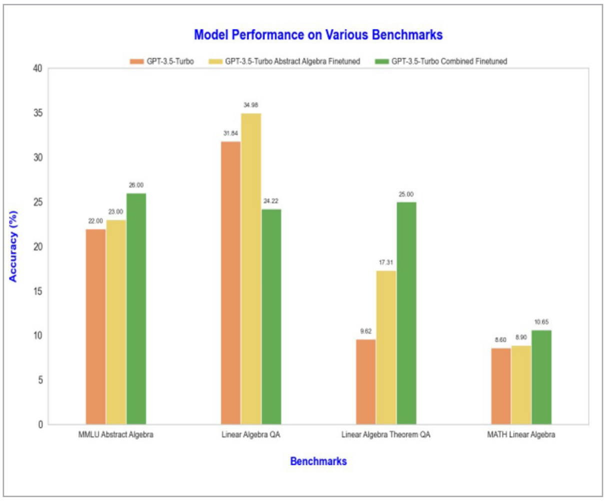
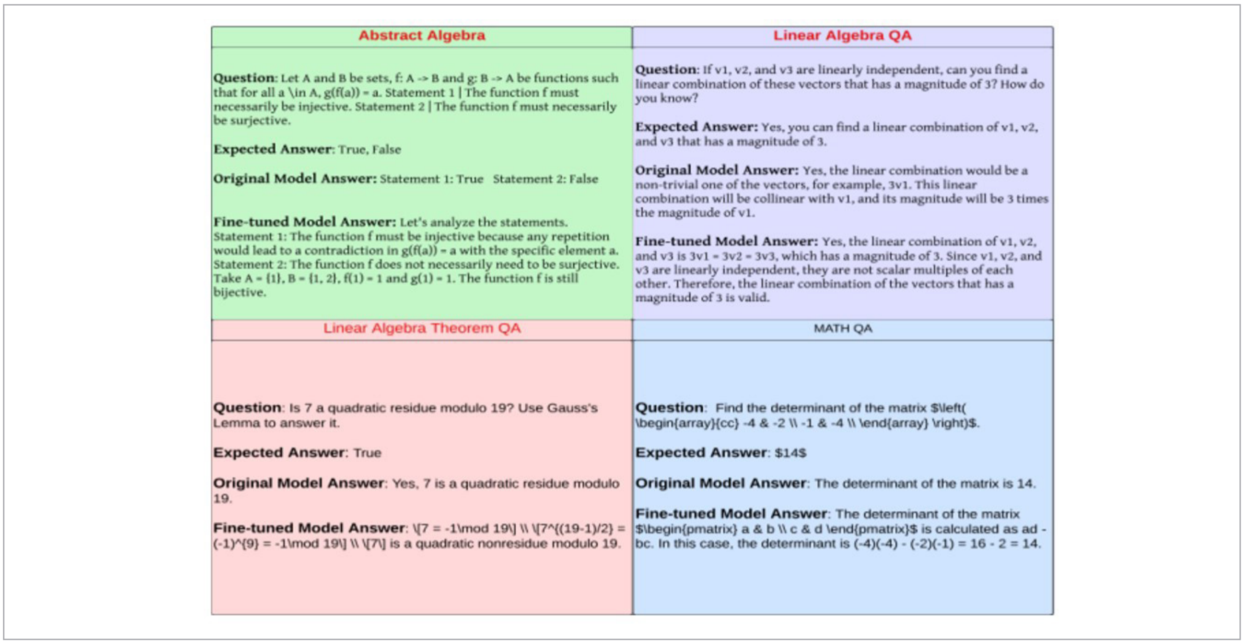


Figure 8

The outputs from original and fine-tuned GPT-3.5-Turbo models on benchmarks. Although the original model could generate correct answers sometimes, fine-tuned models could provide specific reasoning process and better explanations as our fine-tuned data describes



Abstract Algebra	Linear Algebra QA
Question: Let A and B be sets, f: A -> B and g: B -> A be functions such that for all a \in A, g(f(a)) = a. Statement 1 | The function f must necessarily be injective. Statement 2 | The function f must necessarily be surjective. **Expected Answer**: True, False **Original Model Answer:** Statement 1: True Statement 2: False **Fine-tuned Model Answer:** Let's analyze the statements. Statement 1: The function f must be injective because any repetition would lead to a contradiction in g(f(a)) = a with the specific element a. Statement 2: The function f does not necessarily need to be surjective. Take A = {1}, B = {1, 2}, f(1) = 1 and g(1) = 1. The function f is still bijective.	**Question**: If v1, v2, and v3 are linearly independent, can you find a linear combination of these vectors that has a magnitude of 3? How do you know? **Expected Answer:** Yes, you can find a linear combination of v1, v2, and v3 that has a magnitude of 3. **Original Model Answer:** Yes, the linear combination would be a non-trivial one of the vectors, for example, 3v1. This linear combination will be collinear with v1, and its magnitude will be 3 times the magnitude of v1. **Fine-tuned Model Answer:** Yes, the linear combination of v1, v2, and v3 is 3v1 = 3v2 = 3v3, which has a magnitude of 3. Since v1, v2, and v3 are linearly independent, they are not scalar multiples of each other. Therefore, the linear combination of the vectors that has a magnitude of 3 is valid.
Linear Algebra Theorem QA	MATH QA
Question: Is 7 a quadratic residue modulo 19? Use Gauss's Lemma to answer it. **Expected Answer**: True **Original Model Answer**: Yes, 7 is a quadratic residue modulo 19. **Fine-tuned Model Answer**: \[7 = -1\mod 19\] \\ \[7^{(19-1)/2} = (-1)^{9} = -1\mod 19\] \\ \[7\] is a quadratic nonresidue modulo 19.	**Question**: Find the determinant of the matrix $\left(\begin{array}{cc} -4 & -2 \\ -1 & -4 \\ \end{array} \right)$. **Expected Answer**: 14 **Original Model Answer**: The determinant of the matrix is 14. **Fine-tuned Model Answer**: The determinant of the matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is calculated as ad - bc. In this case, the determinant is (-4)(-4) - (-2)(-1) = 16 - 2 = 14.

had an astonishing advancement in Linear Algebra QA benchmark, which surpassed the performance of fine-tuned model on the combined dataset.

Interestingly, we also observed that fine-tuning GPT-3.5 Turbo model on abstract algebra datasets resulted in a notable improvement in accuracy on linear algebra benchmarks, particularly in linear algebra theorem QA. One possible explanation is that the abstract algebra dataset provides the model with a foundational understanding of mathematical structures and concepts that correspond to linear algebra, specifically, vector spaces could be regarded as a group. This overlap in foundational knowledge likely enhanced the mathematical inference ability of model in linear algebra tasks. Furthermore, this observation suggests that our fine-tuned LLM possess generalization capabilities to comprehensively capture the logical correlations across different mathematical areas.

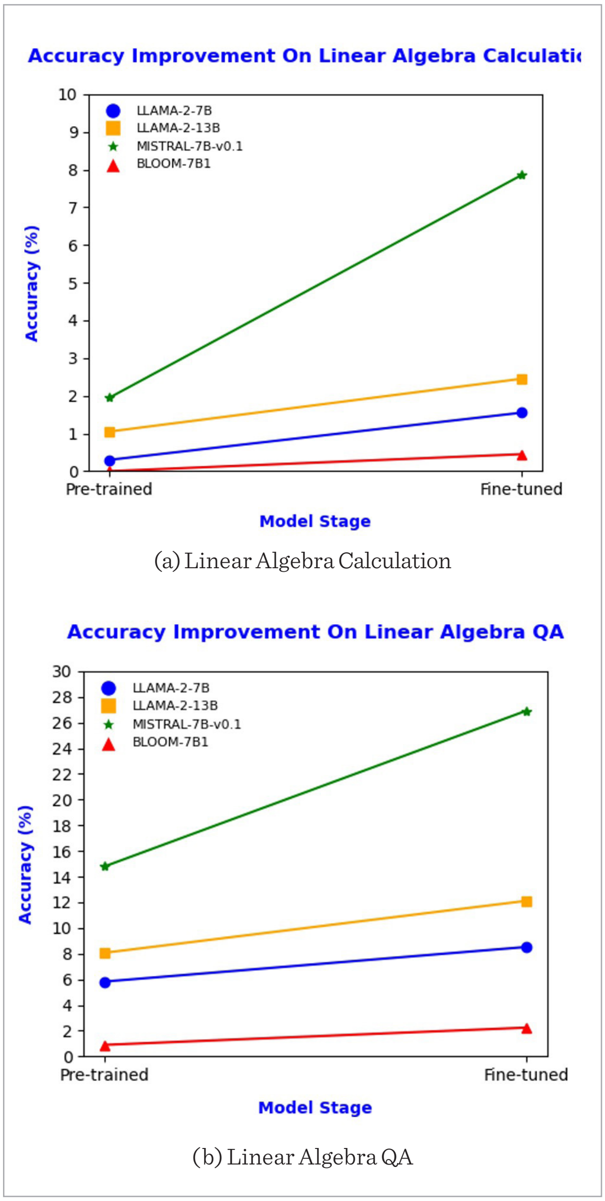
Subsequently, we fine-tuned the SLMs and evaluate their performance on Linear Algebra Calculation and Linear Algebra QA benchmarks which demonstrated reasonable improvements in mathematical reasoning ability as Figure 10 showed. As shown in Figure 10a and Figure 10b, we observed that the Mistral-7B-v0.1 [22] model exhibited best improvements of accuracy on both benchmarks after fine-tuning. Its superior performance might be attributed to its advanced architectures of transformers and attention mechanisms as we mentioned in Section 4.2, and its modification of FlashAttention [10] and xFormers [35] makes training procedures faster.

5. Costs

In our experiments, fine-tuning GPT-3.5-Turbo through OpenAI's API infrastructure had a cost of $5.53 based to its token-based pricing. In contrast, fine-tuning Llama-2 SLMs through Hugging Face's Autotrain platform required only $0.96 and 32 minutes for the 7B model, and $3.15 and 105 minutes for the 13B model, which is more cost-effective than AlpaGasus [5]. Similarly, fine-tuning Mistral-7B-v0.1 cost $1.05 and took 31 minutes, while fine-tuning Bloom 7B1 cost $1.05 and took 35 minutes. Notably, according to Figure 11, Mistral-7B-v0.1 is the most fine-tuning effective model due to its remarkable performance on benchmarks with similar costs and fine-tuning time of LLama-2-7B and Bloom 7B1 models.

Figure 10

The alternation of accuracy of SLMs on Linear Algebra Calculation and Linear Algebra QA benchmark after fine-tuning on our datasets in Section 3



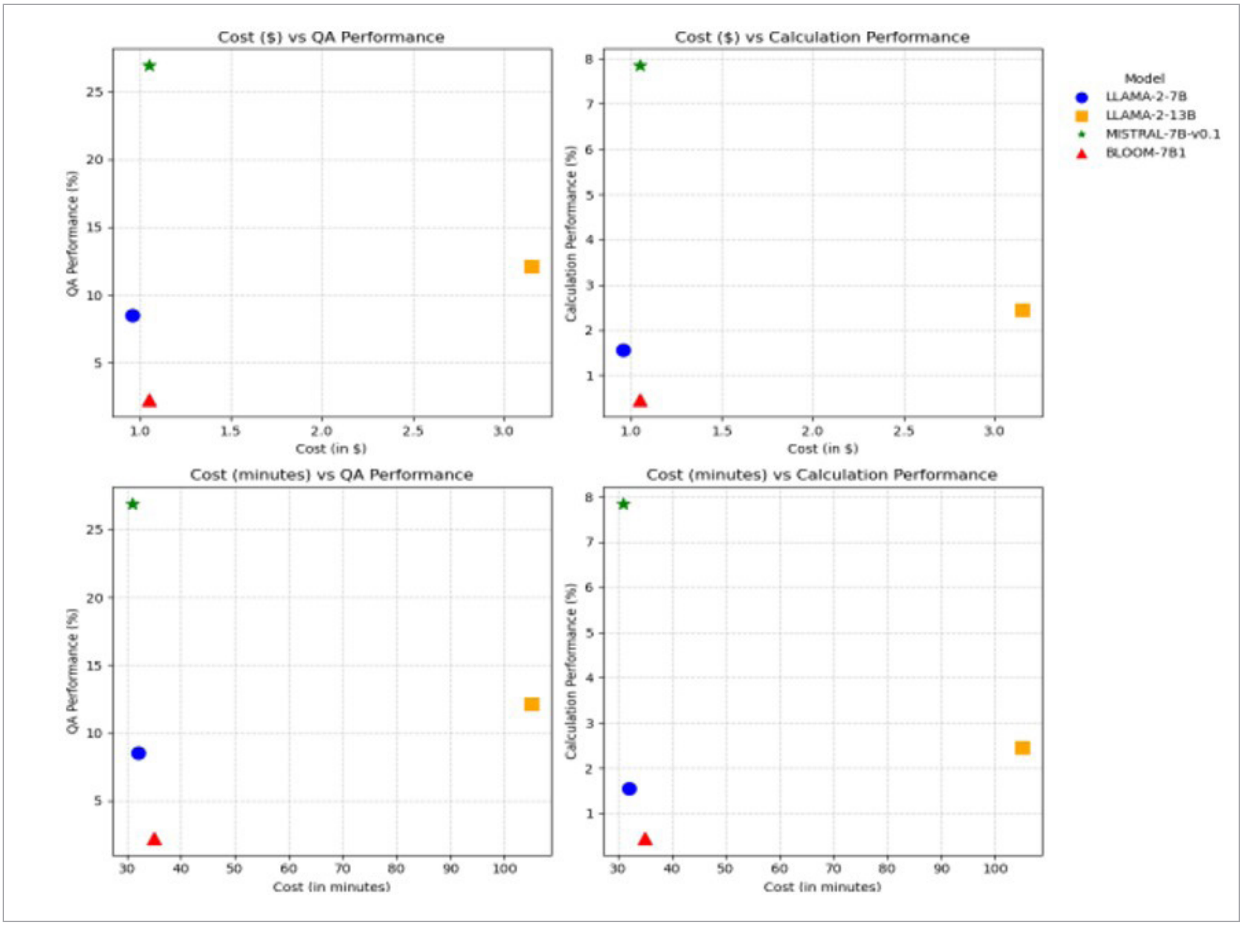
(a) Linear Algebra Calculation

(b) Linear Algebra QA

The low costs of our fine-tuning procedures for SLMs are mainly attributed to efficient application of LoRA [19] which significantly reduced the computational burden of fine-tuning. This highlights how individuals could leverage our method through Autotrain to affordably design and customize language models for their own purposes.

Figure 11

Performance of SLMs and their performance on different benchmarks



6. Discussion

Despite the proven effectiveness of synthetic data for fine-tuning language models in linear algebra and abstract algebra, in order to leverage broad generalization scope of these language models, future works could focus on two approaches: (1) scaling up synthetic datasets by integrating more diverse mathematical domains including topology, calculus, geometry, and number theory. The diverse datasets could likely enhance the generalization capabilities of our fine-tuned models in mathematical reasoning. (2) investigating more advanced base language models such as Falcon-7B and Llama2-70B to assess their ability to solve complex mathematical questions and validate our findings that advanced models can achieve better performance at lower costs.

Both pre-trained open-source language models and their fine-tuned versions are readily available on Huggingface, which offers two user-friendly deployment approaches for individuals without technical background. The first option is to directly deploy models through cooperative cloud platforms, such as Amazon SageMaker or Azure ML, which provides users with optimized CUDA-based environments for running language models. The second approach is to load the model locally using the Transformers library, which is well-suited for users with compatible hardware. Moreover, our fine-tuned mathematical language models maintain potential to be a preview tool to assist senior high school students preparing for undergraduate mathematics courses.

While synthetic data provides convenience in fine-tuning models, the black-box nature of language models introduce uncertainties in data generation process, which has raised critical concerns about data bias, model transparency, and potential impact on education. Since different models in Gretel.ai were pretrained on diverse datasets, the generated mathematical data may inherently contain the biases from the corresponding training data which potentially caused the degradation of models' performance on mathematical reasoning tasks during the fine-tuning process. Furthermore, as a closed-source platform, it is challenging to track the comprehensive parameters of pre-configured models in Gretel.ai, which makes users difficult to explore diversity and variability of synthetic data through tuning more hyperparameters besides T, K, and P. Due to inherent data biases and the lack of model transparency, although mathematical abilities of language models have improved with fine-tuning on synthetic data, they might unintentionally provide incorrect solutions. Therefore, users should apply the answers provided by these models with caution and are encouraged to perform cross-validation of knowledge with responsibility.

7. Conclusion

In conclusion, our method provides a feasible approach to effectively fine-tune mathematical QA language models using synthetic data which yielded notable improvements in algebra calculations and theorems across various language models. Considering the trade-off between cost and performance in fine-tuning, selecting an appropriate pretrained model is crucial to achieve practical usability, and advanced pre-trained SLMs tend to have superior performance after fine-tuning, while requiring less costs and time. It indicates that synthetic data could be an effective and efficient resource for enhancing the mathematical reasoning capabilities of language models, and our method offers individuals a versatile choice to deploy their own fine-tuning tasks.

Beyond the application of synthetic data in enhancing mathematical ability of language models, our findings contributed to the broader AI/ML community in three aspects: (1) synthetic data generation could be extended to other fields such as chemistry and physics for various fine-tuning tasks. (2) it alleviates the difficulty of acquiring labeled data to mitigate the risk of underfitting when training models with limited datasets. (3) the relatively low cost of synthetic data allows AI/ML practitioner to channel funds and time into model architectures and application designs for products.

Data sharing agreement

Our fine-tuned SLMs are available at https://huggingface.co/Charlie-Han-01, and project page is available at https://github.com/DinoZeyu/LLM-Research.git.

Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Funding

The author(s) received no financial support for the research.

Reference

1. Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. ArXiv, 2023, abs/2305.13245. https://doi.org/10.18653/v1/2023.emnlp-main.298
2. Berglund, M., van der Merwe, B. Formalizing BPE Tokenization. Electronic Proceedings in Theoretical Computer Science, 2023, 388, 16-27. https://doi.org/10.4204/EPTCS.388.4
3. BigScience. BLOOM-7B1: BigScience Large Open-Science Open-Access Multilingual Language Model. Hugging Face, 2023. https://huggingface.co/bigscience/bloom-7b1
4. Brown, T. B., et al. Language Models Are Few-Shot Learners. ArXiv, 2020, abs/2005.14165. https://api.semanticscholar.org/CorpusID:218971783
5. Chen, L., Wang, Y., Liu, J. AlpaGasus: Training a Better Alpaca with Fewer Data. ArXiv, 2023, abs/2307.08701

6. Chen, W., Chen, Z., Chen, Z. TheoremQA: A Theorem-Driven Question Answering Dataset. ArXiv, 2023, abs/2305.12524. https://doi.org/10.18653/v1/2023.emnlp-main.489

7. Chowdhery, A., et al. PaLM: Scaling Language Modeling with Pathways. Journal of Machine Learning Research, 2022, 24, 240:1-240:113. https://api.semanticscholar.org/CorpusID:247951931

8. Chowdhery, A., et al. PaLM: Scaling Language Modeling with Pathways. ArXiv, 2022, abs/2204.02311. https://arxiv.org/abs/2204.02311

9. Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C., Schulman, J. Training Verifiers to Solve Math Word Problems. ArXiv, 2021, abs/2110.14168

10. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention With IO-Awareness. ArXiv, 2022, abs/2205.14135.

11. Dettmers, T., Lewis, M., Belkada, Y., Zettlemoyer, L. 8-bit Optimizers via Block-wise Quantization. ArXiv, 2022, abs/2110.02861. https://arxiv.org/abs/2110.02861

12. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. ArXiv, 2023, abs/2305.14314

13. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. ArXiv, 2019, abs/1810.04805. https://arxiv.org/abs/1810.04805

14. Ding, B., Liu, X., Li, J., Zhang, Y., Zhang, H. Data Augmentation Using Large Language Models: Data Perspectives, Learning Paradigms, and Challenges. ArXiv, 2024, abs/2403.02990. https://arxiv.org/abs/2403.02990

15. Dubey, A., Ren, M., Schilling, J., Dey, S., Katabi, D., Liu, M. Y., LeCun, Y. The LLaMA 3 Herd of Models. ArXiv, 2024, abs/2407.21783. https://arxiv.org/abs/2407.21783

16. Hassani, A., Walton, S., Shi, H. Neighborhood Attention Transformer. ArXiv, 2023, abs/2204.07143. https://arxiv.org/abs/2204.07143

17. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J. Aligning AI With Shared Human Values. Proceedings of the International Conference on Learning Representations (ICLR), 2021.

18. Hendrycks, D., Karamcheti, S., Burns, C., Mazeika, M., Tang, E., Song, D., Steinhardt, J. Measuring Mathematical Problem Solving with the MATH Dataset. NeurIPS, 2021.

19. Hu, J. E., Shen, T., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. ArXiv, 2021, abs/2106.09685. https://arxiv.org/abs/2106.09685

20. Isik, B., Khan, N., Akata, Z., Van der Smagt, P. Scaling Laws for Downstream Task Performance of Large Language Models. ArXiv, 2024, abs/2402.04177. https://arxiv.org/abs/2402.04177

21. Yue, X., Wang, C., Zhang, Z., Liu, J., Sun, M. MAmmoTH: Building Math Generalist Models Through Hybrid Instruction Tuning. ArXiv, 2023, abs/2309.05653. https://arxiv.org/abs/2309.05653

22. Jiang, A. Q., Xu, J., Zhang, Z., Wang, Y., Lin, H. Mistral 7B. ArXiv, 2023, abs/2310.06825

23. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q., Chen, J. TinyBERT: Distilling BERT for Natural Language Understanding. ArXiv, 2019, abs/1909.10351. https://arxiv.org/abs/1909.10351

24. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Amodei, D. Scaling Laws for Neural Language Models. ArXiv, 2020, abs/2001.08361. https://arxiv.org/abs/2001.08361

25. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., Iwasawa, Y. Large Language Models Are Zero-Shot Reasoners. ArXiv, 2023, abs/2205.11916. https://arxiv.org/abs/2205.11916

26. Kwon, W., Kim, K., Kim, Y., Oh, A., Kang, U. Efficient Memory Management for Large Language Model Serving with PagedAttention. Proceedings of the 29th Symposium on Operating Systems Principles, 2023. https://doi.org/10.1145/3600006.3613165

27. Lambda Labs. Demystifying GPT-3. Lambda Labs Blog, June 2020. https://lambdalabs.com/blog/demystifying-gpt-3

28. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. ArXiv, 2019, abs/1909.11942. https://arxiv.org/abs/1909.11942

29. Li, M., Zhang, J., Yang, Y., Sun, X., Liu, J., Wu, W. From Quantity to Quality: Boosting LLM Performance with Self-Guided Data Selection for Instruction Tuning. ArXiv, 2023, abs/2308.12032. https://arxiv.org/abs/2308.12032

30. Li, Q., Wang, J., Zhang, L., Shen, Z., Zhou, H., Yang, M. LaFFi: Leveraging Hybrid Natural Language Feedback for Fine-Tuning Language Models. ArXiv, 2023, abs/2401.00907. https://arxiv.org/abs/2401.00907

31. Liang, Z., Yu, W., Rajpurohit, T., Clark, P., Zhang, X., Kaylan, A. Let GPT Be a Math Tutor: Teaching Math Word

Problem Solvers with Customized Exercise Generation. ArXiv, 2023, abs/2305.14386. https://doi.org/10.18653/v1/2023.emnlp-main.889

32. Likhi2003. Linear Algebra QA Dataset. Hugging Face, 2024. https://huggingface.co/datasets/Likhi2003/linearalgebra_QA
33. Ling, W., Yogatama, D., Dyer, C., Blunsom, P. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. Annual Meeting of the Association for Computational Linguistics, 2017. https://doi.org/10.18653/v1/P17-1015
34. Luo, H., Sun, Q., Xu, C., Zhao, P., Lou, J., Tao, C., Geng, X., Lin, Q., Chen, S., Tang, Y., Zhang, D. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. ArXiv, 2023, abs/2308.09583. https://arxiv.org/abs/2308.09583
35. Meta AI Research. xFormers: A Modular and Fast Transformer Library. GitHub, 2023. https://facebookresearch.github.io/xformers/
36. Meta AI. Meta LLaMA 3.1: Open Foundation Models for Research and Commercial Use. Meta AI Blog, 2023. https://ai.meta.com/blog/meta-llama-3-1/
37. Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient Estimation of Word Representations in Vector Space. International Conference on Learning Representations, 2013. https://api.semanticscholar.org/CorpusID:5959482
38. OpenAI. GPT-3.5 Turbo Fine-Tuning and API Updates. OpenAI Blog, 2023. https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates/
39. OpenAI. GPT-4 Technical Report. ArXiv, 2024, abs/2303.08774. https://arxiv.org/abs/2303.08774
40. OpenAI. GPT-4o Mini: Advancing Cost-Efficient Intelligence. OpenAI Blog, 2024. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/
41. OpenAI. OpenAI o1-mini: Advancing Cost-Efficient Reasoning. OpenAI Blog, 2024. https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/
42. Peng, S., Yuan, K., Gao, L., Tang, Z. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. ArXiv, 2021, abs/2105.00377. https://api.semanticscholar.org/CorpusID:233481495
43. Pennington, J., Socher, R., Manning, C. D. GloVe: Global Vectors for Word Representation. Conference on Empirical Methods in Natural Language Processing, 2014. https://doi.org/10.3115/v1/D14-1162
44. Press, O., Smith, N. A., Lewis, M. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. ArXiv, 2022, abs/2108.12409. https://arxiv.org/abs/2108.12409
45. Radford, A.,Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. Language Models Are Unsupervised Multitask Learners. OpenAI Blog, 2019, 1.8, 9.
46. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. Exploring the Limits of Transfer Learning With a Unified Text-to-Text Transformer. Journal of Machine Learning Research, 2019, 21, 140:1-140:67. https://api.semanticscholar.org/CorpusID:204838007
47. Sanh, V., Debut, L., Chaumond, J., Wolf, T. DistilBERT: A Distilled Version of BERT-Smaller, Faster, Cheaper, and Lighter. ArXiv, 2020, abs/1910.01108. https://arxiv.org/abs/1910.01108
48. Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., Catanzaro, B. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. ArXiv, 2020, abs/1909.08053. https://arxiv.org/abs/1909.08053
49. Smith, S., et al. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. ArXiv, 2022, abs/2201.11990. https://arxiv.org/abs/2201.11990
50. Stanford Alpaca. An Instruction-Following LLaMA Model. GitHub, 2023. https://github.com/tatsu-lab/stanford_alpaca
51. Touvron, H., et al. LLaMA 2: Open Foundation and Fine-Tuned Chat Models. ArXiv, 2023, abs/2307.09288. https://api.semanticscholar.org/CorpusID:259950998
52. Touvron, H., et al. LLaMA: Open and Efficient Foundation Language Models. ArXiv, 2023, abs/2302.13971. https://arxiv.org/abs/2302.13971
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention Is All You Need. Neural Information Processing Systems, 2017. https://api.semanticscholar.org/CorpusID:13756489
54. Wang, Y., Xu, C., Sun, Q., Hu, H., Tao, C., Geng, X., Jiang, D. PromDA: Prompt-Based Data Augmentation for Low-Resource NLU Tasks. Annual Meeting of the Association for Computational Linguistics, 2022. https://doi.org/10.18653/v1/2022.acl-long.292
55. Wang, W., Wei, F., Dong, L., Bao, H., Yang, B., Zhou, M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transform-

ers. ArXiv, 2020, abs/2002.10957. https://arxiv.org/abs/2002.10957

56. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. ArXiv, 2023, abs/2201.11903. https://arxiv.org/abs/2201.11903

57. Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., Le, Q. V. Fine-Tuned Language Models Are Zero-Shot Learners. ArXiv, 2022, abs/2109.01652. https://arxiv.org/abs/2109.01652

58. Xia, Y., Kim, J., Chen, Y., Ye, H., Kundu, S., Hao, C., Talati, N. Understanding the Performance and Estimating the Cost of LLM Fine-Tuning. ArXiv, 2024, abs/2408.04693. https://arxiv.org/abs/2408.04693

59. Zanibbi, R., Aizawa, A., Kohlhase, M., Ounis, I., Topic, G., Davila, K. NTCIR-12 MathIR Task Overview. NTCIR Conference on Evaluation of Information Access Technologies, 2016. https://api.semanticscholar.org/CorpusID:9102694

60. Zhang, P., Zeng, G., Wang, T., Lu, W. TinyLlama: An Open-Source Small Language Model. ArXiv, 2024, abs/2401.02385. https://arxiv.org/abs/2401.02385