

Synthetic Data Enhances Mathematical Reasoning of Language Models

Zeyu Han^{1,*}

First Name Last Name²

December 9, 2024

Abstract

Current large language models (LLMs) training involves extensive training data and computing resources to handle in 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 an 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. 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>.

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

1 Introduction

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 [1, 2] to Transformer based encoder and decoder autoregressive models [3, 4, 5, 6, 7, 8], the flourishing progress of LLMs depends on appearance of Transformer structure [9], innovation of effective finetuning algorithms and techniques [10, 11, 12, 13], and the gradually increasing diversity and scale of training data.

In order to improve the ability of LLMs, [14] 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 [15]. 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 [5] as an example, it is trained on V100 GPU high-bandwidth clusters with mixed datasets composed of CommonCrawl [4] and WebText [16] totaling nearly 430 billion tokens and its training expenses exceed \$4.6 million [17].

Meanwhile, data quality has became 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 [5]. Similarly, training LLMs with adequate AI generated NLP feedback data and efficient parametric fine-tuning technique LoRA [12] could effectively improve the

performance of QA task in low-data scenarios [18].

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. On the other hand, individuals could train small mathematical language models to fulfill personal demands.

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 [19]. 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 [20]; (ii) Ensure the diversity and accuracy of data, in other words, data quality [21]. 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 effective BERT-based SLMs:DistilBERT [22], ALBERT [23], TinyBERT [24], and MiniLM [25]. 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 [26, 27]. Fine-tuning LLMs with a fewer well-filtered dataset, i.e., instruction fine-tuning data [21], is a practical approach that can enable the model to achieve SoTA performance on various tasks [28].

Notably, [29] 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, [30] proposed that using prompt-based data augmentation could ensure the quality of synthetic data which would increase the natural language understanding of models.

Prior researches have 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

2.1 LLM Overview

In recent years, general multimodal LLMs have an impressive development in many areas of NLP such as BERT series [3], T-5 [4], Megatron-Turing NLG [31], LLaMA series [8], ChatGPT series [32], PaLM [33], and these models have outstanding performance on many NLP tasks as shown in Figure 1. Meanwhile, these models also have corresponding SLMs such as GPT-4o mini [34], LLama-2-7B/13B [7], and TinyLlama [35]. 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.

LLM Development



Figure 1: Recent years representative LLMs which achieve SoTA performance during their periods. **BERT** [3] initially used bidirectional Transformer architectures. **T-5** [4] proposed to covert NLP tasks into a text-to-text form. **GPT-3** [5] performed excellent effectiveness of few-shot and zero-shot learners and made a wide influence. **PaLM** [33] strongly validated the Scaling Laws in LLMs pre-training. **GPT-4** [32] and **LLAMA3** [36] are the most popular and well-known LLMs present days due to their high performance.

2.2 Math-Related Models

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.

AlpaGasus: Developed by [28], the AlpaGasus model represents a feasible technique that utilizing powerful LLMs to mitigate the performance reduction of Alpaca [37] 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 [38] 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 [39, 40, 41] has a significantly improvement compared to other open source models such as WizardMath [42].

MathBERT: Unlike other models, MathBERT [43] 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 [44] benchmark and remarkable precision and recall on TopicMath-100K [43] benchmark. It performed outstanding results on mathematical information retrieval, formula topic classification and formula headline generation downstream tasks.

o1-mini: In September 14th 2024, OPENAI released the o1-MINI model [45] which made an 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 o1 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

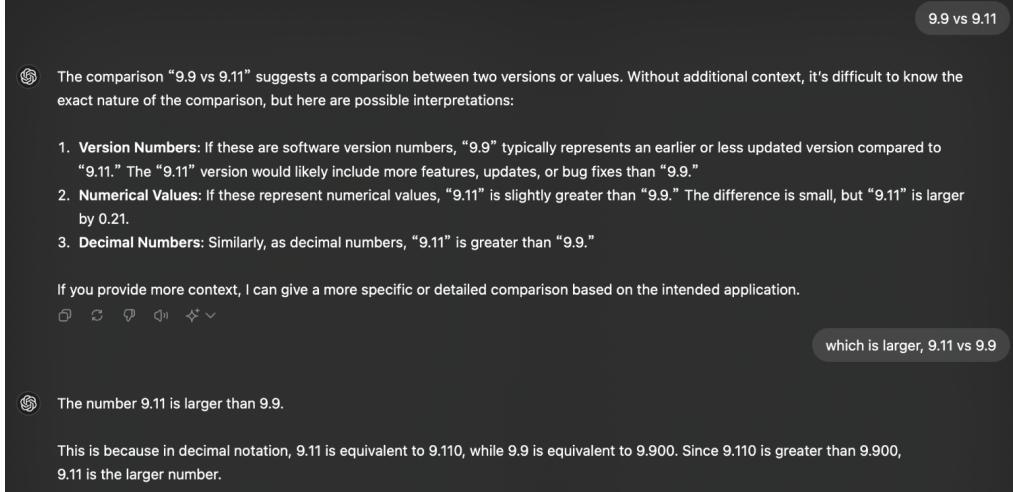


Figure 2: ChatGPT 4o sometimes may have confusion to figure out which number is larger, 9.11 or 9.9.

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 AlpaGagus [28] 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 [38] and MathBERT [43].

3 Data Description

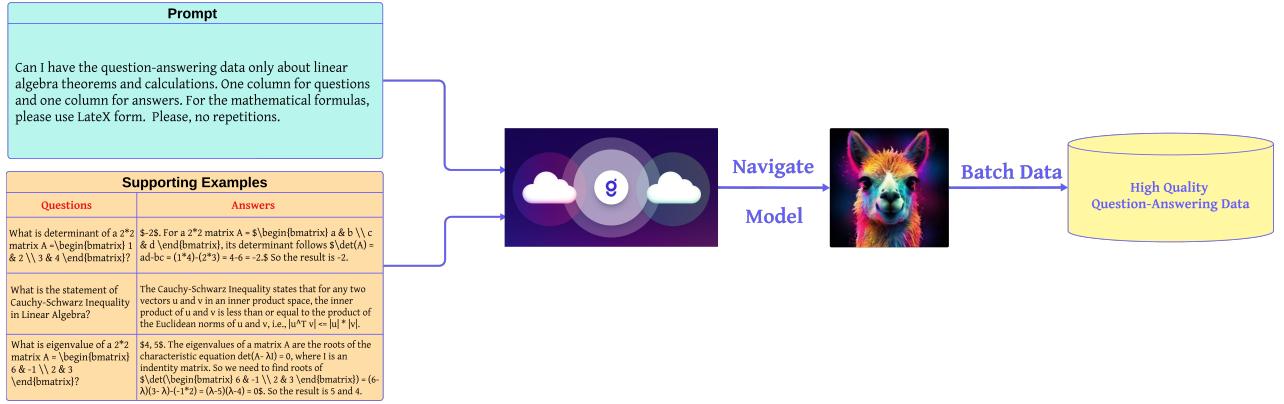


Figure 3: Data generation platform GRETEL.AI. We provide the prompt and sample data for GRETEL.AI cloud to create navigator model. According to navigator, the platform chooses the GRETEL-LLAMA-3.1-8B-INSTRUCT model to batch synthetic linear algebra data.

3.1 Fine-Tuning Data

The dataset 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.

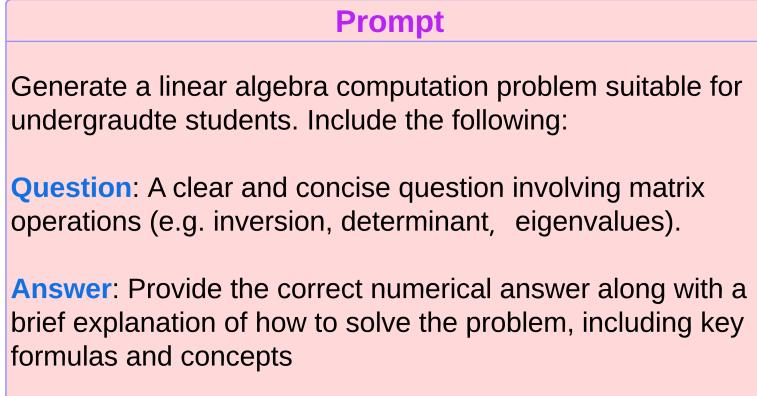


Figure 4: Prompt to synthesize Linear Algebra Computation QA Data

For the data generation process, as shown in Figure 3, we initially refined our requirements for synthetic data according to an elaborate prompt, and examples covering theorems and calculations pertinent to the specific mathematical field. Then, the GRETEL.AI platform generated 100 rows of tabular data with 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 in generated data. Subsequently, the platform leveraged existing prompt and data to construct a navigator model capable of selecting appropriate fine-tuned models and generating data in batches as required. The linear algebra data was generated from Gretel-LLAMA-3.1-8B [46] and abstract algebra data was generated from Gretel GPT-3.5 Turbo [47]. In addition, we have standardized the mathematical formulas into LaTeX format to guarantee consistency.

Nevertheless, we observed that the linear algebra dataset contains few computational problems and corresponding solutions. Although language models possess zero-shot learning capabilities [5], the lack 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 Fig 4, which could be considered as data augmentation [48].

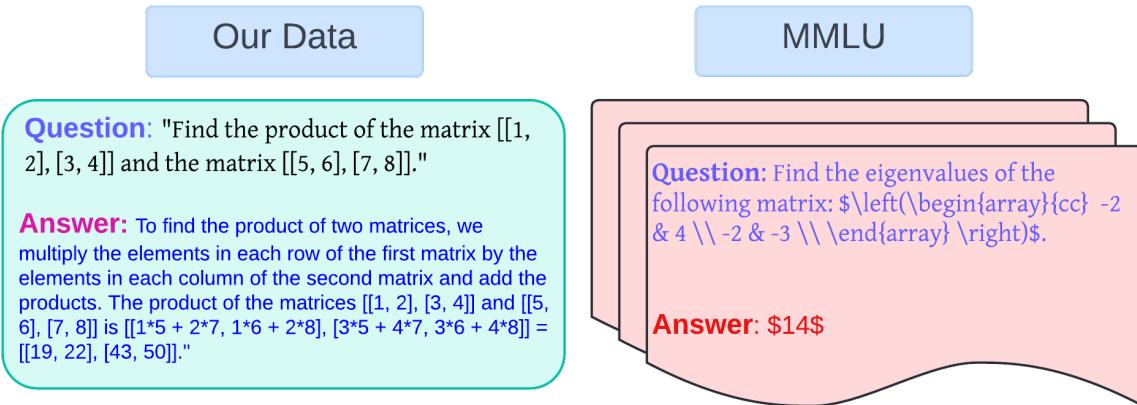


Figure 5: Data difference between two datasets. In our dataset, we included the process of solving problems, which is similar to chain-of-thought [49] to get outputs compared to MMLU.

In contrast to prior research [28, 38, 29], 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 the GRETEL.AI provides all users free 1.5 million characters usage per month.

3.2 Benchmarks

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	[50]	52	Evaluation
MATH	[39]	2.0k	Evaluation
Linear Algebra QA	[51]	223	Evaluation
Partial MMLU	[52]	101	Evaluation

Table 1: Overview of datasets and benchmarks used in the experiments.

In order to examine the feasibility of our fine-tuning method, we chose widely used mathematical benchmarks and take 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.

TheoremQA [50] 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 [39] 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 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 [51] 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 [52] is a comprehensive benchmark covering 57 subjects across STEM to evaluate models’ performance under zero-shot or few-shot settings. In 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

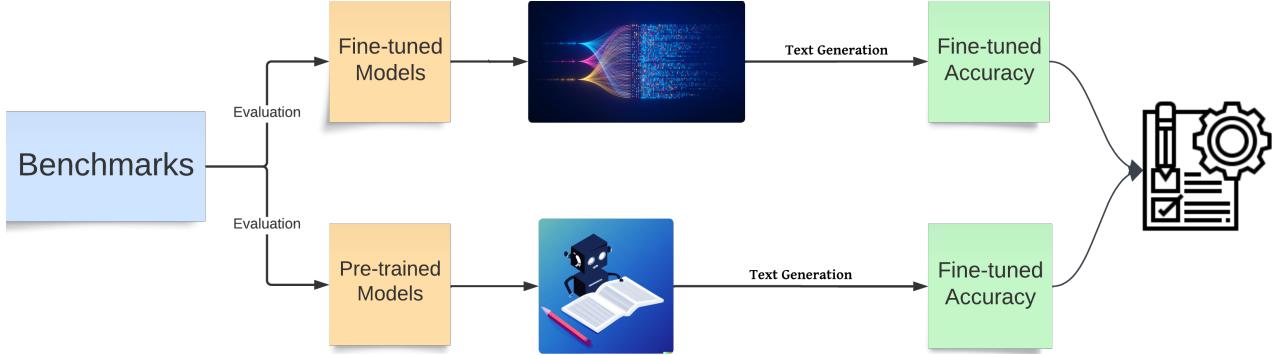


Figure 6: Workflows of our experiment

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

- Accuracy: The primary metric in mathematical QA tasks about calculations and proofs. While some linear algebra and abstract algebra problems necessitate theoretical proofs, evaluating the reasonability of these answers often involves assessing the accuracy of generated answers and logical steps of the proof.
- 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 [8] are open-sourced auto-regressive models developed by Meta with 2 trillion pretraining tokens, 4092 context lengths, and over 100K fine-tuning data.

MISTRAL-7B-v0.1 [53] is an open-sourced model developed by Mistral AI with the usage of Grouped-Query Attention [54], Sliding-Window Attention [55], and Byte-fallback BPE tokenizer [56] techniques to enhance the efficiency and performance of the model on many NLP tasks.

BLOOM-7B1 [57] is a multilingual SLM developed by BigScience which is a decoder-only model modified from MEGATRON-LM GPT2 [58] and was trained using 8-bit optimizers [59] and ALIBI positional encodings [60].

GPT-3.5-TURBO [47] 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 benchmark

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.

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

Table 2: Accuracy of Language Models on Algebra Benchmarks.

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 [14], 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 user-friendly interface and cost-effectiveness which makes it accessible for people without coding experience.

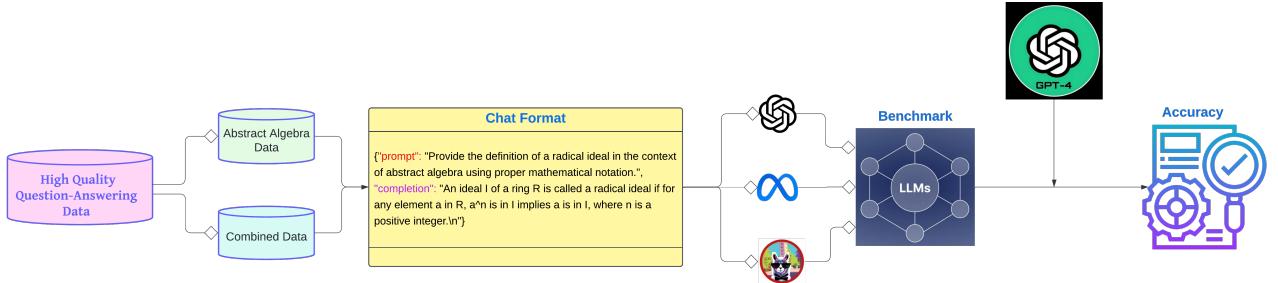


Figure 7: After obtaining the fine-tuning data, we separated them into two subsets: Abstract Algebra and Combined Dataset. Then we used different datasets to fine-tune models and took accuracy as our metric for evaluation according to GPT-4 model.

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.
- 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.

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

4.5 Results

Abstract Algebra	Linear Algebra QA
<p>Question: Let A and B be sets, $f: A \rightarrow B$ and $g: B \rightarrow A$ 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.</p> <p>Expected Answer: True, False</p> <p>Original Model Answer: Statement 1: True Statement 2: False</p> <p>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.</p>	<p>Question: If v_1, v_2, and v_3 are linearly independent, can you find a linear combination of these vectors that has a magnitude of 3? How do you know?</p> <p>Expected Answer: Yes, you can find a linear combination of v_1, v_2, and v_3 that has a magnitude of 3.</p> <p>Original Model Answer: Yes, the linear combination would be a non-trivial one of the vectors, for example, $3v_1$. This linear combination will be collinear with v_1, and its magnitude will be 3 times the magnitude of v_1.</p> <p>Fine-tuned Model Answer: Yes, the linear combination of v_1, v_2, and v_3 is $3v_1 + 3v_2 - 3v_3$, which has a magnitude of 3. Since v_1, v_2, and v_3 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.</p>
Linear Algebra Theorem QA	MATH QA
<p>Question: Is 7 a quadratic residue modulo 19? Use Gauss's Lemma to answer it.</p> <p>Expected Answer: True</p> <p>Original Model Answer: Yes, 7 is a quadratic residue modulo 19.</p> <p>Fine-tuned Model Answer: $\lfloor 7 = -1 \bmod 19 \rfloor \wedge \lfloor 7^{\frac{(19-1)}{2}} = (-1)^9 = -1 \bmod 19 \rfloor \wedge \lfloor 7 \rfloor$ is a quadratic nonresidue modulo 19.</p>	<p>Question: Find the determinant of the matrix $\begin{pmatrix} cc & -4 & -2 \\ -1 & c & -4 \end{pmatrix}$.</p> <p>Expected Answer: \$14\$</p> <p>Original Model Answer: The determinant of the matrix is 14.</p> <p>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 $(-4)(-4) - (-2)(-1) = 16 - 2 = 14$.</p>

Figure 8: The outputs from original and fine-tuned GPT-3.5-TURBO models on benchmarks. Although original model could generate correct answers sometimes, fine-tuned model could provide specific reasoning process and better explanations as our fine-tuned data describes.

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 [49]. 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 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 yielded a notable improvement in accuracy on linear algebra benchmarks, particularly in linear algebra theorem QA as Figure 9 showed. One of the possible reasons is that the abstract algebra dataset provides the model with a foundational understanding

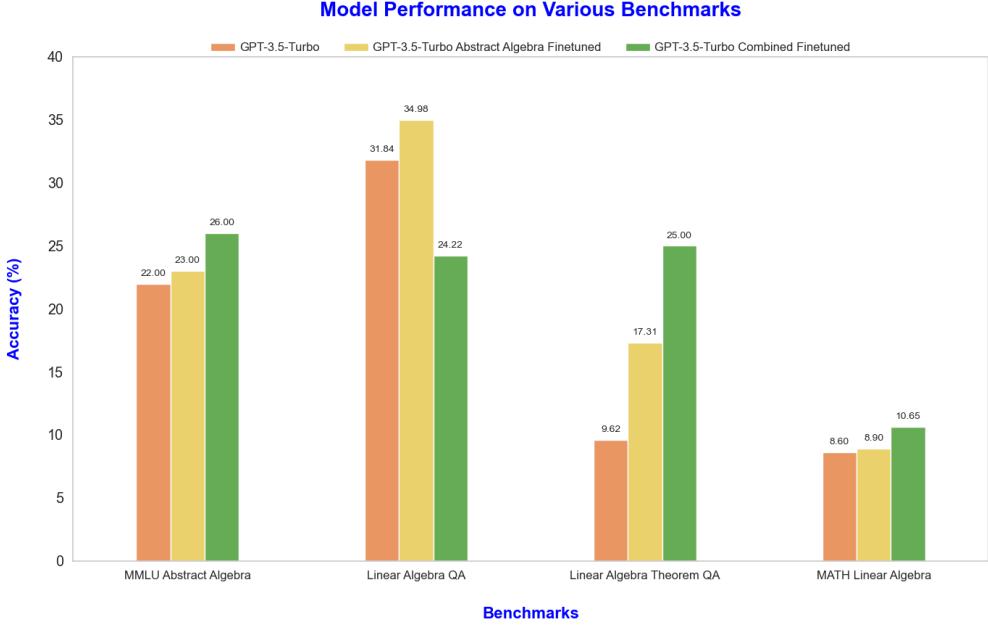


Figure 9: Performance of GPT-3.5-TURBO and its fine-tuned models across various datasets and benchmarks.

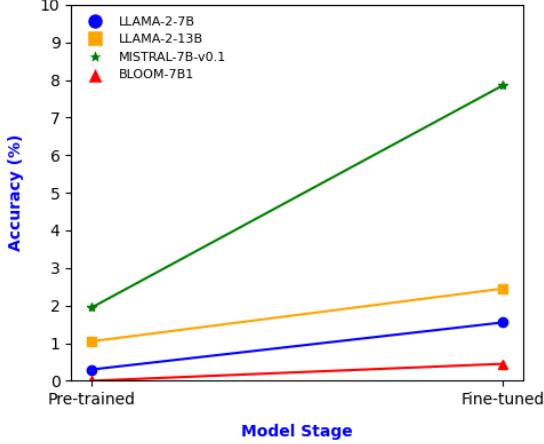
of mathematical structures and concepts that correspond to linear algebra, specifically, vector spaces could be regarded as a group. These two areas exhibit significant overlap in foundational knowledge which enhanced the mathematical inference ability of model in linear algebra tasks.

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 [53] 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 [11] and xFormers [61] makes training procedures faster.

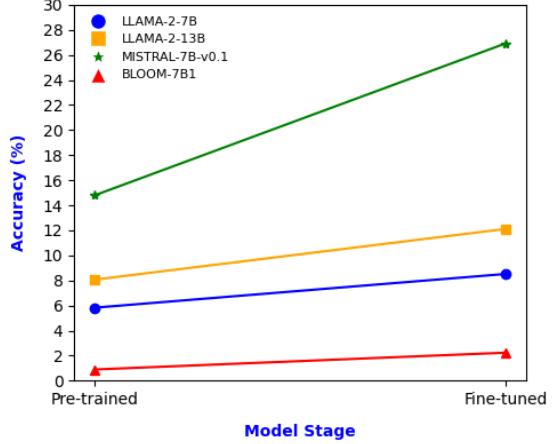
5 Cost

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 [28]. 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.

The low costs of our fine-tuning procedures for SLMs is mainly attributed to efficient application of LoRA [12] 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.

Accuracy Improvement On Linear Algebra Calculation

(a) Linear Algebra Calculation

Accuracy Improvement On Linear Algebra QA

(b) Linear Algebra QA

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.

6 Conclusion

In conclusion, our method provides a feasible approach to fine-tune mathematical QA language models using synthetic data. Our cost-effective fine-tuning method yielded notable improvements in linear algebra calculations and formulas across various models, including LLMs (e.g., GPT-3) and SLMs (e.g., LLAMA-2-7B and MISTRAL-7B-v0.1). Considering the trade-off between cost and performance in fine-tuning, selecting an appropriate pretrained model is crucial to achieve practical usability. Advanced pre-trained SLM tend to have superior performance after fine-tuning, while requiring less costs and time. Our study indicates that synthetic data could be an effective and efficient approach for enhancing the mathematical reasoning capabilities of language models, and our method offers individuals a potentially optimal choice to deploy their own fine-tuning tasks.

7 Limitations

Our study represents a basic exploration into fine-tuning language models with synthetic mathematical data, demonstrating the potential of this approach for enhancing mathematical reasoning and developing personalized models. However, several avenues warrant future investigation to improve the generalization and performance of models. On the one hand, future work should explore incorporating a broader range of mathematical data types, including topology, calculus, and geometry, to enhance the generalization capabilities of the fine-tuned models. On the other hand, exploring more advanced base language models, such as FALCON-7B, could provide greater capacity for solving complex mathematical questions.

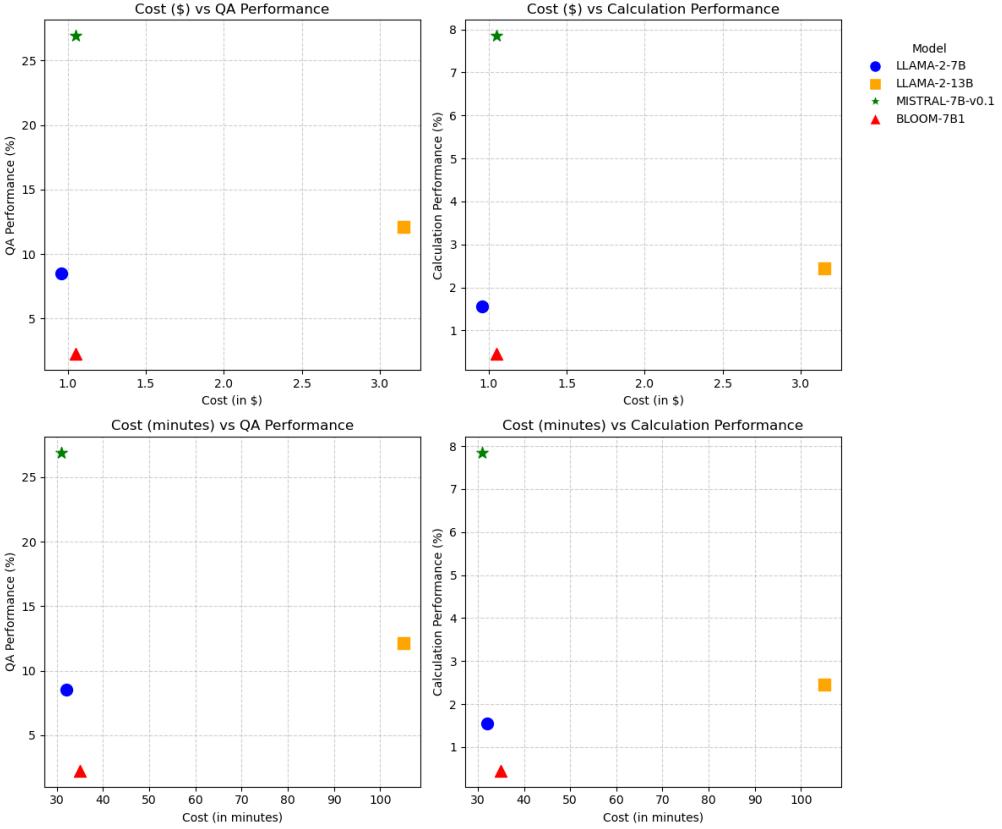


Figure 11: Performance of SLMs and their costs on different benchmarks.

References

- [1] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *International Conference on Learning Representations*. 2013. URL: <https://api.semanticscholar.org/CorpusID:5959482>.
- [2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Conference on Empirical Methods in Natural Language Processing*. 2014. URL: <https://api.semanticscholar.org/CorpusID:1957433>.
- [3] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: [1810.04805 \[cs.CL\]](https://arxiv.org/abs/1810.04805). URL: <https://arxiv.org/abs/1810.04805>.
- [4] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *J. Mach. Learn. Res.* 21 (2019), 140:1–140:67. URL: <https://api.semanticscholar.org/CorpusID:204838007>.
- [5] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: *ArXiv* abs/2005.14165 (2020). URL: <https://api.semanticscholar.org/CorpusID:218971783>.
- [6] Aakanksha Chowdhery et al. “PaLM: Scaling Language Modeling with Pathways”. In: *J. Mach. Learn. Res.* 24 (2022), 240:1–240:113. URL: <https://api.semanticscholar.org/CorpusID:247951931>.
- [7] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: [2302.13971 \[cs.CL\]](https://arxiv.org/abs/2302.13971). URL: <https://arxiv.org/abs/2302.13971>.
- [8] Hugo Touvron et al. “Llama 2: Open Foundation and Fine-Tuned Chat Models”. In: *ArXiv* abs/2307.09288 (2023). URL: <https://api.semanticscholar.org/CorpusID:259950998>.
- [9] Ashish Vaswani et al. “Attention is All you Need”. In: *Neural Information Processing Systems*. 2017. URL: <https://api.semanticscholar.org/CorpusID:13756489>.

- [10] Woosuk Kwon et al. “Efficient Memory Management for Large Language Model Serving with PagedAttention”. In: *Proceedings of the 29th Symposium on Operating Systems Principles* (2023). URL: <https://api.semanticscholar.org/CorpusID:261697361>.
- [11] Tri Dao et al. “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness”. In: *ArXiv* abs/2205.14135 (2022). URL: <https://api.semanticscholar.org/CorpusID:249151871>.
- [12] J. Edward Hu et al. “LoRA: Low-Rank Adaptation of Large Language Models”. In: *ArXiv* abs/2106.09685 (2021). URL: <https://api.semanticscholar.org/CorpusID:235458009>.
- [13] Tim Dettmers et al. “QLoRA: Efficient Finetuning of Quantized LLMs”. In: *ArXiv* abs/2305.14314 (2023). URL: <https://api.semanticscholar.org/CorpusID:258841328>.
- [14] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: [2001.08361 \[cs.LG\]](https://arxiv.org/abs/2001.08361). URL: <https://arxiv.org/abs/2001.08361>.
- [15] Yuchen Xia et al. *Understanding the Performance and Estimating the Cost of LLM Fine-Tuning*. 2024. arXiv: [2408.04693 \[cs.CL\]](https://arxiv.org/abs/2408.04693). URL: <https://arxiv.org/abs/2408.04693>.
- [16] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.
- [17] Lambda Labs. *Demystifying GPT-3*. June 2020. URL: <https://lambdalabs.com/blog/demystifying-gpt-3>.
- [18] Qianxi Li et al. *LaFFi: Leveraging Hybrid Natural Language Feedback for Fine-tuning Language Models*. 2023. arXiv: [2401.00907 \[cs.LG\]](https://arxiv.org/abs/2401.00907). URL: <https://arxiv.org/abs/2401.00907>.
- [19] Berivan Isik et al. *Scaling Laws for Downstream Task Performance of Large Language Models*. 2024. arXiv: [2402.04177 \[cs.CL\]](https://arxiv.org/abs/2402.04177). URL: <https://arxiv.org/abs/2402.04177>.
- [20] Jason Wei et al. *Finetuned Language Models Are Zero-Shot Learners*. 2022. arXiv: [2109.01652 \[cs.CL\]](https://arxiv.org/abs/2109.01652). URL: <https://arxiv.org/abs/2109.01652>.
- [21] Ming Li et al. “From Quantity to Quality: Boosting LLM Performance with Self-Guided Data Selection for Instruction Tuning”. In: *ArXiv* abs/2308.12032 (2023). URL: <https://api.semanticscholar.org/CorpusID:261076515>.
- [22] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: [1910.01108 \[cs.CL\]](https://arxiv.org/abs/1910.01108). URL: <https://arxiv.org/abs/1910.01108>.
- [23] Zhenzhong Lan et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *CoRR* abs/1909.11942 (2019). arXiv: [1909.11942](https://arxiv.org/abs/1909.11942). URL: [http://arxiv.org/abs/1909.11942](https://arxiv.org/abs/1909.11942).
- [24] Xiaoqi Jiao et al. “Tinybert: Distilling bert for natural language understanding”. In: *arXiv preprint arXiv:1909.10351* (2019).
- [25] Wenhui Wang et al. *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*. 2020. arXiv: [2002.10957 \[cs.CL\]](https://arxiv.org/abs/2002.10957).
- [26] Wenhui Wang et al. *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*. 2020. arXiv: [2002.10957 \[cs.CL\]](https://arxiv.org/abs/2002.10957). URL: <https://arxiv.org/abs/2002.10957>.
- [27] Zhenwen Liang et al. “Let GPT be a Math Tutor: Teaching Math Word Problem Solvers with Customized Exercise Generation”. In: *ArXiv* abs/2305.14386 (2023). URL: <https://api.semanticscholar.org/CorpusID:258865450>.
- [28] Lichang Chen et al. “AlpaGasus: Training A Better Alpaca with Fewer Data”. In: *ArXiv* abs/2307.08701 (2023). URL: <https://api.semanticscholar.org/CorpusID:259937133>.
- [29] Takeshi Kojima et al. *Large Language Models are Zero-Shot Reasoners*. 2023. arXiv: [2205.11916 \[cs.CL\]](https://arxiv.org/abs/2205.11916). URL: <https://arxiv.org/abs/2205.11916>.
- [30] Yufei Wang et al. “PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks”. In: *Annual Meeting of the Association for Computational Linguistics*. 2022. URL: <https://api.semanticscholar.org/CorpusID:247155039>.

- [31] Shaden Smith et al. *Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model*. 2022. arXiv: [2201.11990 \[cs.CL\]](https://arxiv.org/abs/2201.11990). URL: <https://arxiv.org/abs/2201.11990>.
- [32] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: [2303.08774 \[cs.CL\]](https://arxiv.org/abs/2303.08774). URL: <https://arxiv.org/abs/2303.08774>.
- [33] Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways*. 2022. arXiv: [2204.02311 \[cs.CL\]](https://arxiv.org/abs/2204.02311). URL: <https://arxiv.org/abs/2204.02311>.
- [34] OpenAI. *GPT-4o Mini: Advancing Cost-Efficient Intelligence*. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>. Accessed: September 4, 2024. 2024.
- [35] Peiyuan Zhang et al. *TinyLlama: An Open-Source Small Language Model*. 2024. arXiv: [2401.02385 \[cs.CL\]](https://arxiv.org/abs/2401.02385). URL: <https://arxiv.org/abs/2401.02385>.
- [36] Abhimanyu Dubey et al. *The Llama 3 Herd of Models*. 2024. arXiv: [2407.21783 \[cs.AI\]](https://arxiv.org/abs/2407.21783). URL: <https://arxiv.org/abs/2407.21783>.
- [37] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023.
- [38] Xiang Yue et al. “MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning”. In: *ArXiv* abs/2309.05653 (2023). URL: <https://api.semanticscholar.org/CorpusID:261696697>.
- [39] Dan Hendrycks et al. “Measuring Mathematical Problem Solving With the MATH Dataset”. In: *NeurIPS* (2021).
- [40] Karl Cobbe et al. “Training Verifiers to Solve Math Word Problems”. In: *ArXiv* abs/2110.14168 (2021). URL: <https://api.semanticscholar.org/CorpusID:239998651>.
- [41] Wang Ling et al. “Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems”. In: *Annual Meeting of the Association for Computational Linguistics*. 2017. URL: <https://api.semanticscholar.org/CorpusID:12777818>.
- [42] Haipeng Luo et al. *WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct*. 2023. arXiv: [2308.09583 \[cs.CL\]](https://arxiv.org/abs/2308.09583). URL: <https://arxiv.org/abs/2308.09583>.
- [43] Shuai Peng et al. “MathBERT: A Pre-Trained Model for Mathematical Formula Understanding”. In: *ArXiv* abs/2105.00377 (2021). URL: <https://api.semanticscholar.org/CorpusID:233481495>.
- [44] Richard Zanibbi et al. “NTCIR-12 MathIR Task Overview”. In: *NTCIR Conference on Evaluation of Information Access Technologies*. 2016. URL: <https://api.semanticscholar.org/CorpusID:9102694>.
- [45] OpenAI. “OpenAI o1-mini: Advancing Cost-Efficient Reasoning”. In: (2024). URL: <https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/>.
- [46] Meta AI. *Meta LLaMA 3.1: Open Foundation Models for Research and Commercial Use*. <https://ai.meta.com/blog/meta-llama-3-1/>. Accessed: 2024-09-03. Aug. 2023.
- [47] OpenAI. *GPT-3.5 Turbo fine-tuning and API updates*. <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates/>. Accessed: 2024-09-03. Aug. 2023.
- [48] Bosheng Ding et al. *Data Augmentation using Large Language Models: Data Perspectives, Learning Paradigms and Challenges*. 2024. arXiv: [2403.02990 \[cs.CL\]](https://arxiv.org/abs/2403.02990). URL: <https://arxiv.org/abs/2403.02990>.
- [49] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: [2201.11903 \[cs.CL\]](https://arxiv.org/abs/2201.11903). URL: <https://arxiv.org/abs/2201.11903>.
- [50] Wenhui Chen et al. “TheoremQA: A Theorem-driven Question Answering dataset”. In: *ArXiv* abs/2305.12524 (2023). URL: <https://api.semanticscholar.org/CorpusID:258833200>.
- [51] Likhi2003. *Linear Algebra QA Dataset*. https://huggingface.co/datasets/Likhi2003/linearalgebra_QA. Accessed: 2024-11-24. 2024.
- [52] Dan Hendrycks et al. “Aligning AI With Shared Human Values”. In: *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

- [53] Albert Q. Jiang et al. *Mistral 7B*. 2023. arXiv: [2310.06825 \[cs.CL\]](https://arxiv.org/abs/2310.06825). URL: <https://arxiv.org/abs/2310.06825>.
- [54] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*. 2023. arXiv: [2305.13245 \[cs.CL\]](https://arxiv.org/abs/2305.13245). URL: <https://arxiv.org/abs/2305.13245>.
- [55] Ali Hassani et al. *Neighborhood Attention Transformer*. 2023. arXiv: [2204.07143 \[cs.CV\]](https://arxiv.org/abs/2204.07143). URL: <https://arxiv.org/abs/2204.07143>.
- [56] Martin Berglund and Brink van der Merwe. “Formalizing BPE Tokenization”. In: *Electronic Proceedings in Theoretical Computer Science* 388 (Sept. 2023), pp. 16–27. ISSN: 2075-2180. DOI: [10.4204/eptcs.388.4](https://dx.doi.org/10.4204/EPTCS.388.4). URL: <http://dx.doi.org/10.4204/EPTCS.388.4>.
- [57] BigScience. *BLOOM-7B1: BigScience Large Open-science Open-access Multilingual Language Model*. <https://huggingface.co/bigscience/bloom-7b1>. Accessed: 2024-11-17. 2023.
- [58] Mohammad Shoeybi et al. *Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*. 2020. arXiv: [1909.08053 \[cs.CL\]](https://arxiv.org/abs/1909.08053). URL: <https://arxiv.org/abs/1909.08053>.
- [59] Tim Dettmers et al. *8-bit Optimizers via Block-wise Quantization*. 2022. arXiv: [2110.02861 \[cs.LG\]](https://arxiv.org/abs/2110.02861). URL: <https://arxiv.org/abs/2110.02861>.
- [60] Ofir Press, Noah A. Smith, and Mike Lewis. *Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation*. 2022. arXiv: [2108.12409 \[cs.CL\]](https://arxiv.org/abs/2108.12409). URL: <https://arxiv.org/abs/2108.12409>.
- [61] Meta AI Research. *xFormers: A Modular and Fast Transformer Library*. <https://facebookresearch.github.io/xformers/>. Accessed: 2024-11-17. 2023.