Autonomous Data Selection with Zero-shot Generative Classifiers for Mathematical Texts

Yifan Zhang[1*], **Yifan Luo**[1*], **Yang Yuan**[1,2†], **Andrew C Yao**[1,2†]
[1]Tsinghua University, [2]Shanghai Qi Zhi Institute
zhangyif21@mails.tsinghua.edu.cn, luoyf24@mails.tsinghua.edu.cn
yuanyang@tsinghua.edu.cn, andrewcyao@tsinghua.edu.cn

Abstract

We present *Autonomous Data Selection* (**AutoDS**), a method that leverages base language models themselves as zero-shot "generative classifiers" to automatically curate high-quality mathematical texts. Unlike prior approaches that require human annotations or training a dedicated data filter, AutoDS relies solely on a model's logits to determine whether a given passage is mathematically informative and educational. By integrating AutoDS into a continual pretraining pipeline, we substantially boost downstream performance on challenging math benchmarks (MATH, GSM8K, and BBH) while using far fewer tokens than previous methods. Empirically, our approach achieves roughly a twofold improvement in pretraining token efficiency over strong baselines, underscoring the potential of self-directed data selection in enhancing mathematical reasoning. We release our curated AutoMathText dataset to facilitate future research in automated domain-specific data curation[§]. The AutoMathText dataset is available at https://huggingface.co/datasets/math-ai/AutoMathText.

1 Introduction

Language models (LMs) have witnessed tremendous advancements, becoming increasingly adept at natural language understanding, generation, and reasoning (Devlin et al., 2019; Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023; Anil et al., 2023). Yet, integrating *domain-specific* knowledge into these models remains a critical and challenging frontier (Lewkowycz et al., 2022; Azerbayev et al., 2024). Mathematical reasoning, in particular, demands specialized expertise: texts often contain symbolic formulas, multi-step derivations, and rigorous proof structures that differ considerably from conventional language tasks (Hendrycks et al., 2021; Paster et al., 2024; Wang et al., 2023b). Despite the growing enthusiasm for building LMs with robust mathematical proficiency, the field continues to face a scarcity of well-curated and high-quality mathematical corpora, underscoring the urgent need for innovative approaches to create and refine domain-specific training data.

Recent efforts have begun to address this gap. For instance, Gunasekar et al. (2023) and Li et al. (2023) demonstrated the utility of large LMs (e.g., GPT-4) to appraise the educational value of code snippets in the Stack dataset (Kocetkov et al., 2022), subsequently training a traditional classifier (e.g., random forest) for data filtering. While these approaches represent a pivotal step toward more judicious data curation, they typically produce only *discrete* labels (e.g., "good" vs. "bad"), discarding the finer granularity of data quality. In mathematical contexts, subtle nuances matter: a dataset entry with an "educational value" of 0.95 should arguably be treated differently from one at 0.001. Relying solely on binary classification can thus limit the efficiency and precision of the training pipeline.

A promising alternative is to assign *continuous* real-valued scores to each data point, thereby enabling the model to focus selectively on the most informative texts. However, constructing such a continuous scoring system poses nontrivial challenges. Large language models often struggle with generating reliable numerical values or sampling consistently from intricate distributions (Hopkins et al., 2023; Hu et al., 2024). Drawing inspiration from the Direct Preference Optimization (DPO) framework (Rafailov et al., 2023), we propose a simpler yet effective solution: leveraging the model's own logits associated with targeted tokens (e.g., "YES" vs. "NO") to produce a quantitative score function. This approach avoids costly labeling efforts and bypasses the need for training an additional classifier

[*]Equal contribution.
[†]Corresponding authors.
[§]The code is available at https://github.com/yifanzhang-pro/AutoMathText.

Findings of the Association for Computational Linguistics: ACL 2025, pages 4168–4189

on human-annotated data.

Concretely, we introduce Autonomous Data Selection (**AutoDS**), which uses zero-shot meta-prompts to evaluate the quality of mathematical texts for continual pretraining. Instead of relying on aligned or fine-tuned models, we take a strong *base* model and prompt it with two yes/no questions assessing (1) the level of "mathematical intelligence" in the text, and (2) its utility for future math learning. From the resulting logits on "YES" and "NO," we compute a single real-valued LM-SCORE that captures the text's educational value. This enables a more fine-grained assessment than binary filtering approaches (Li et al., 2023; Paster et al., 2024), thus amplifying token efficiency by selectively training on the most instructive samples.

Another distinguishing factor of our method is its ability to *autonomously* curate data: no separate human-annotated corpus or reward model is needed. Techniques like supervised fine-tuning (SFT) (Radford et al., 2019), Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), or specialized preference modeling (Rafailov et al., 2023) are not required. By directly applying a softmax-based score on the base model's logits, AUTODS orchestrates a form of active, self-directed learning, where the model itself identifies and harnesses the best materials for continual pretraining. This paves the way for a more *dynamic* and *scalable* data selection pipeline, especially relevant for highly specialized fields like mathematics.

Empirically, we show that continually pretraining language models on this *auto-curated* dataset yields substantial gains on mathematics benchmarks, such as MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and BIG-Bench Hard (BBH) (Suzgun et al., 2022). Remarkably, these improvements come with far fewer tokens compared to previous continual pretraining works, effectively boosting training efficiency by roughly a factor of two. Figure 1 previews these performance trends on the Mistral-7B model (Jiang et al., 2023), underscoring the efficacy of our data selection method.

Our key contributions are three-fold:

- We propose a straightforward, zero-shot generative classifier framework that uses *logits*-based scoring to automatically filter large-scale mathematical data. It circumvents the need for supervised or human feedback signals while retaining fine-grained control over data quality.
- We assemble and release a carefully curated dataset, *AutoMathText*, drawn from multiple high-value sources (e.g., OpenWebMath, arXiv, Algebraic Stack). It addresses the scarcity of domain-specific mathematical corpora essential for training more powerful LMs.
- Through extensive evaluations, we demonstrate that LMs continually pretrained with AUTODS achieve significantly higher accuracy on mathematical tasks, surpassing binary-based filtering methods and achieving a $2\times$ increase in pretraining token efficiency.

2 Language Models as Zero-shot Generative Classifiers

Recent advances in large language models (LLMs) have demonstrated remarkable potential for complex reasoning and decision-making (Wei et al., 2022; Bubeck et al., 2023). Building on these capabilities, we propose leveraging *base* language models in a zero-shot fashion to verify whether candidate documents possess the mathematical rigor and educational utility necessary for continual pretraining. This approach goes beyond conventional few-shot paradigms (Brown et al., 2020), which require task-specific prompt engineering or model fine-tuning, by directly harnessing the LLMs' inherent capacity to assess textual content without reliance on human annotations.

Generative Classifiers. The centerpiece of our AUTODS framework is a scoring function using LLMs as *generative classifiers* for quantifying the model's propensity to affirm or deny the mathematical value of a given piece of text. Specifically, we examine the logits associated with "YES" and "NO" when the model is prompted with two diagnostic questions (e.g., *Is this text mathematically intelligent? Is it educational for future math learning?*). Let $\mathrm{logit}(\mathrm{YES})$ and $\mathrm{logit}(\mathrm{NO})$ denote the output logits of the model for these tokens. We define the zero-shot LM-Score as follows:

$$\begin{aligned}&\text{LM-Score}(\cdot)\\ =&\frac{\exp(\mathrm{logit}(\mathrm{YES}))}{\exp(\mathrm{logit}(\mathrm{YES}))+\exp(\mathrm{logit}(\mathrm{NO}))}.\end{aligned} \quad (1)$$

Here, a higher value indicates the model's stronger inclination to judge the text as mathematically

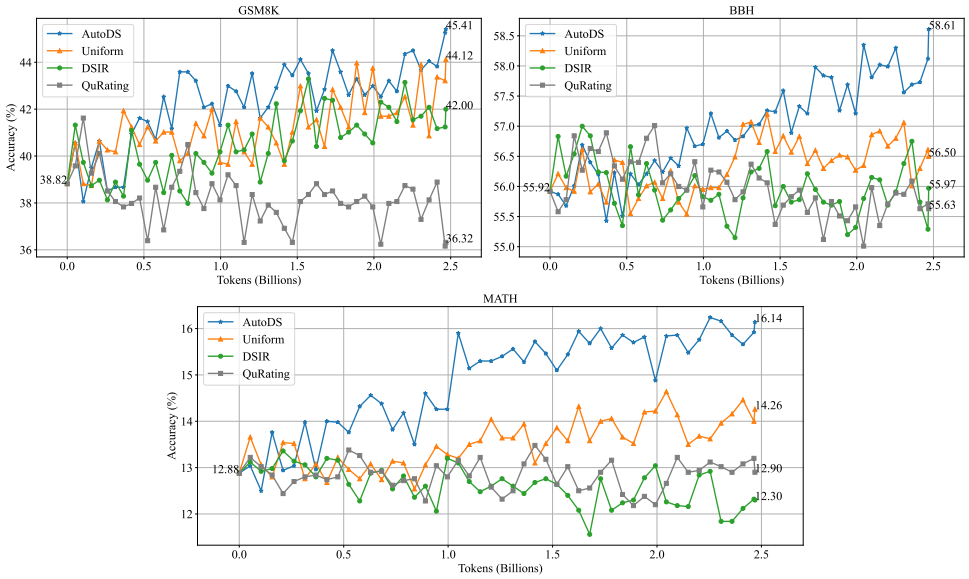

Figure 1: Visualization of Mistral-7B's performances of continual pretrained models with different data selection methods on GSM8K (Hendrycks et al., 2021), BIG-Bench Hard (BBH) (Suzgun et al., 2022) and MATH (Hendrycks et al., 2021) tasks.

valuable. Notably, this mirrors the Bradley-Terry model from reward modeling in RLHF (Ouyang et al., 2022), yet our method requires no supervised dataset nor explicit preference labels.

Zero-shot Meta-prompts. To elicit these logits in a consistent and interpretable manner, we formulate a concise meta-prompt (Zhang et al., 2023) that asks two questions about each candidate text. As shown in Figure 2, the prompt is presented in a structured format, and the model is directed to respond only with "YES" or "NO." Crucially, we extract the logits from the underlying language model before any additional sampling. This procedure obviates the need for manual filtering or annotated corpora.

Because the meta-prompt poses two questions, we compute the LM-SCORE by multiplying the probabilities corresponding to "YES" for each question:

$$\text{LMScore}(Q_1, Q_2) = \text{LM-Score}(Q_1) \times \text{LM-Score}(Q_2). \quad (2)$$

Thus, a document must be deemed sufficiently positive on *both* dimensions—mathematical intelligence and educational worth—to achieve a high overall score.

Autonomous Continual Pretraining. A critical advantage of our approach is its ease of integration into *continual pretraining* pipelines. Rather than training a secondary classifier or obtaining human labels, the base model itself autonomously selects or discards documents over time. By re-evaluating each new batch of data, the model can dynamically refine its own training corpus, effectively learning "what to learn next." This self-directed mechanism is especially appealing for specialized

"
<system>
You are ChatGPT, equipped with extensive expertise in mathematics and coding, and skilled in complex reasoning and problem-solving. In the following task, I will present a text excerpt from a website. Your role is to evaluate whether this text exhibits mathematical intelligence and if it is suitable for educational purposes in mathematics. Please respond with only YES or NO
</system>
User: {
"url": "{url}",
"text": "{text}"
}
1. Does the text exhibit elements of mathematical intelligence? Respond with YES or NO
2. Is the text suitable for educational purposes for YOURSELF in the field of mathematics? Respond with YES or NO
"
Assistant: 1.

Figure 2: Illustration of our zero-shot meta-prompt designed for AUTODS. The underlying model is instructed to respond only with "YES" or "NO," thereby enabling a direct extraction of logits for each answer.

domains (e.g., mathematics), where human annotations are often scarce, expensive, or unreliable.

Avoiding Human Annotations. Finally, our zero-shot strategy obviates the necessity of extensive labeled datasets or alignment with human preferences (e.g., via RLHF). This decision reflects accumulating evidence suggesting that strong LLMs exhibit competitive (and, in many cases, superior) capacity for domain-specific judgement (Burns et al., 2024). This autonomy is critical for mathematics, where naive keyword-based heuristics (e.g., counting LaTeX symbols) may fail to capture deeper aspects of mathematical reasoning. By leaning on the model's emergent understanding, we thus enable more scalable, cost-efficient data curation, as demonstrated in Figure 3 and Figure 9 in Appendix C.

In summary, our zero-shot generative classification technique exploits a model's intrinsic capacity to rate documents for mathematical utility without any additional training. This paradigm paves the way for *self-supervised* data selection, drastically reducing the need for hand-labeled resources and potentially accelerating the development of LLMs proficient in mathematical reasoning.

3 Autonomous Data Selection with Language Models

Building on the zero-shot verification approach outlined in Section 2, we apply our *LM-Score*-based data selection pipeline to three principal sources of mathematical texts:

1. **OpenWebMath** (Paster et al., 2024): A curated subset of Common Crawl, already filtered for general mathematical content;

2. **arXiv** (from RedPajama) (Weber et al., 2024): Scholarly papers encompassing diverse STEM disciplines;

3. **Algebraic Stack** (Kocetkov et al., 2022; Azerbayev et al., 2024): A specialized subset of GitHub (the "Stack" dataset), featuring code and discussions related to algebraic geometry.

These sources cover a wide range of mathematical domains and difficulty levels, making them well-suited for continual pretraining.

Experiment Details. We process a total of 11.26 M documents, amounting to over 200 GB of data. Following the methodology presented in Section 3, we obtain an *LM-Score* for each document using the Qwen-72B base language model (Bai et al., 2023) and retain documents scoring above specified thresholds. We employ the vLLM inference framework (Kwon et al., 2023) on nodes with A100-80G and A800-80G GPUs. The entire filtering procedure required roughly 750 GPU hours on 4 A100-80G GPUs (i.e., 3000 GPU hours total), including both loading and inference. By contrast, expert manual annotation of 11.26 M documents (at around $1 per document) would cost well over $10 M. Using standard commercial cloud pricing for GPU compute ($2/hour for an A100), our method's budget remains under $10K, drastically reducing labeling cost while avoiding the pitfalls of rule-based or purely keyword-based filtering.

Visualization of Data Composition Examining how the selected data are distributed across different websites and content types provides insight into the quality and variety of the resulting corpus. In Figure 4, we plot a tree map showing the top 30 domains that scored in two different LM-Score ranges. As indicated, `*.stackexchange.com` contributes a substantial share of high-scoring examples, many of which are not yet fully leveraged in other open-source math corpora (Wang et al., 2023b; Liu et al., 2024).

Figure 5 offers a more granular breakdown of the highest-frequency domains and the proportion of documents falling into high-scoring bins (e.g., 0.75–1.00). We observe that many math-intensive websites such as `math.stackexchange.com` and `mathhelpforum.com` have a particularly large share of high-scoring data, underscoring their suitability for enhancing advanced mathematical language modeling.

4 Experiments

In this section, we empirically assess the effectiveness of our proposed AutoDS method in enhancing mathematical reasoning through continual pretraining. We demonstrate that our approach substantially improves performance on several math-focused tasks while using significantly fewer tokens than previous works. We further compare AutoDS against existing baselines and evaluate its broader impact on general reasoning tasks.

"Commutative Property Of Addition. If A is an $n \times m$ matrix and O is a $m \times k$ zero-matrix, then we have: AO = O. Note that AO is the $n \times k$ zero-matrix. ..."
[LM-Score (Q_1, Q_2): 0.946]
[OWMath Classifier Score: 0.767]

"Inequality involving sums with binomial coefficient I am trying to show upper- and lower-bounds on $\frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}\min(i, n-i)$ (where $n \geq 1$) to show that it grows as $\Theta(n)$. The upper-bound is easy to get since $\min(i, n-i) \leq i$ for $i \in \{0, \ldots n\}$ so that $\frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}\min(i, n-i) \leq \frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}i = \frac{n}{2}$. ..." [LM-Score ($Q_1$, Q_2): 0.931]
[OWMath Classifier Score: 0.999]

"The radius of convergence is half the length of the interval of convergence. We noticed that, at least in the case of the geometric series, there was an interval in which it converged, but it didn't converge at the endpoints. Show that the following alternating harmonic series converges: Series of Both Positive and Negative Terms Theorem: Convergence of Absolute Values Implies Convergence If $\sum |a_n|$ converges, then so does $\sum a_n$. Let $f : [1, \infty) \to \mathbb{R}_+$ be a non-negative ... "
[LM-Score (Q_1, Q_2): 0.923]
[OWMath Classifier Score: 0.906]

"# User talk:173.79.37.192 ## March 2009 Welcome to Wikipedia. Although everyone is welcome to make constructive contributions to Wikipedia, at least one of your recent edits, such as the one you made to Reaction time, did not appear to be constructive and has been reverted. Please use the sandbox for any test edits you would like to make, and read the welcome page to learn more about contributing constructively to this encyclopedia. Thank you. Hotcrocodile (talk) 01:33, 11 March 2009 (UTC) If this is a shared IP address, and you didn't make any unconstructive edits, consider creating an account for yourself so you can avoid further irrelevant warnings. ## NAYLA MATTHW [1] [[Media:Example.oggfhf... "
[LM-Score (Q_1, Q_2): 1.58×10^{-5}]
[OWMath Classifier Score: 0.612]

" I've just had one recent comment flag declined on a noisy comment. This comment was a reply to a deleted '+1' comment and said simply: @FrankL Thanks! "
[LM-Score (Q_1, Q_2): 1.21×10^{-5}]
[OWMath Classifier Score: 0.830]

Figure 3: Several examples on selecting web texts. The first example in the left column is from 'track-it.nz', while the second one in the left column is from 'math.stackexchange.com', and the third one in the left column is from 'bwni.pw'. In the right column, the first example is from 'wikipedia.org', and the second one is from 'math.stackexchange.com'. The trained classifier (denoted as OWMath Classifier) used in OpenWebMath (Paster et al., 2024) may mainly focus on how many latex symbols, $ and digits exist in the text, and the examples in the right column show that it may not be very effective.

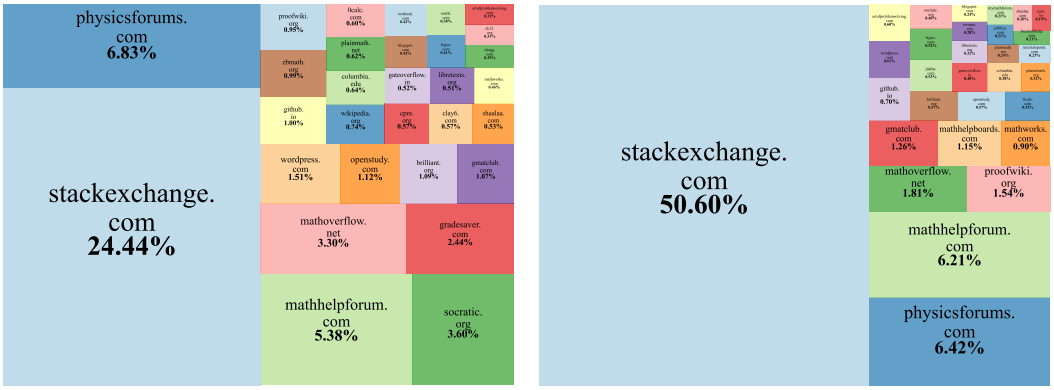

Figure 4: Data composition visualization for the top-30 domains. The left treemap displays documents with LM-Scores of 0.50–1.00, while the right focuses on 0.75–1.00. StackExchange sites form a large proportion of high-score texts, many of which remain underexplored in existing math corpora.

Table 1: MATH test accuracy after continual pretraining and fine-tuning using different data (OpenWebMath and our selected data AutoMathText using method AutoDS).

LM-Score	Type	# Tokens (M)	MATH Acc. (CPT) (%)	MATH Acc. (SFT) (%)
-	Baseline (w/o pretraining)	0	12.88	27.20
-	OpenWebMath	328.9	10.50	26.98
0.75-1.00	**AutoDS**	328.9	**13.68**	**28.06**

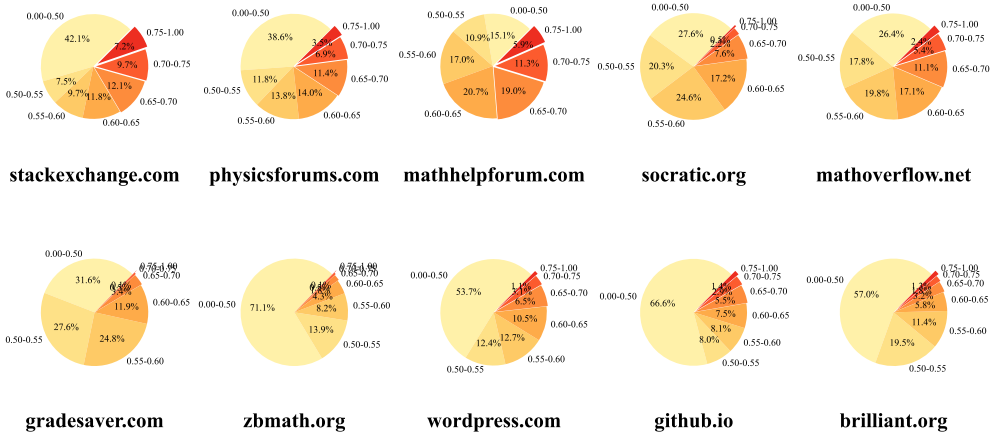

Figure 5: Distribution of LM-Scores among the top-10 domain occurrences, highlighting varying quality levels across sources.

Table 2: Comparison of continual pretrained models using different data selection methods on complex reasoning tasks, showcasing the notable superiority of the AutoDS method.

Model & Selection Method	MATH (5-shot)	GSM8K (5-shot)	BIG-Bench Hard (3-shot)
Gemma-2B Base	10.96	17.29	34.19
+ Uniform (OpenWebMath)	10.16	**18.88**	**36.34**
+ DSIR	5.62	11.90	34.43
+ QuRating	9.76	13.19	31.76
+ AutoDS	**11.02**	**18.88**	34.88
LLaMA2-7B Base	2.94	12.51	39.89
+ Uniform (OpenWebMath)	5.14	19.79	41.53
+ DSIR	2.56	12.51	39.49
+ QuRating	2.90	10.54	39.27
+ AutoDS	**7.74**	**21.99**	**42.76**
Mistral-7B Base	12.88	38.82	55.92
+ Uniform (OpenWebMath)	14.26	44.12	56.50
+ DSIR	12.30	42.00	55.97
+ QuRating	12.90	36.32	55.63
+ AutoDS	**16.14**	**45.41**	**58.61**

4.1 Experiment Details

Base Models. We consider multiple base language models: Gemma-2B (Team et al., 2024), LLaMA2-7B (Touvron et al., 2023), and Mistral-7B (Jiang et al., 2023). These models represent mid-scale LMs frequently used in research and industry. Throughout our experiments, all methods use the same hyperparameters and training schedule for fair comparisons.

Datasets for Continual Pretraining. We continually pretrain each model on selected portions of mathematical text. Our proposed AUTODS filtering (§2–§3) retains only the top-scoring documents (based on LM-Score) from three major sources:

1. OpenWebMath (Paster et al., 2024),
2. arXiv (from RedPajama) (Weber et al., 2024),
3. Algebraic Stack (Kocetkov et al., 2022; Azerbayev et al., 2024).

We focus primarily on the Web subset for this work, applying zero-shot verification via Qwen-72B (Bai et al., 2023) to compute the LM-Scores. In total, we process over 11.26 M documents (200 GB). Documents that exceed specified LM-Score thresholds are included in the final dataset, which we call *AutoMathText*.

Data Selection Baselines. We compare AUTODS with:

1. **Uniform (OpenWebMath):** Uniformly sampled data from OpenWebMath (Paster et al.,

Table 3: Comprehensive comparison of continual pretrained models across diverse reasoning and comprehension tasks. The table is divided into two major sections: world knowledge and reading comprehension.

Model & Selection Method	NQ (5)	$\text{MMLU}_{\text{STEM}}$ (5)	ARC-E (25)	ARC-C (25)	SciQ (2)	LogiQA (2)	BoolQ (0)	Average
Gemma-2B Base	14.88	36.60	77.61	46.50	96.30	25.35	69.54	42.92
+ Uniform (OpenWebMath)	13.80	36.54	77.40	45.39	96.40	26.27	68.35	42.95
+ DSIR	13.27	34.44	77.27	45.82	95.90	23.50	54.92	39.71
+ QuRating	14.32	33.81	77.95	46.76	96.20	24.42	68.53	41.67
+ AutoDS	13.27	36.09	76.81	46.08	96.10	27.19	71.28	**43.16**
LLaMA2-7B Base	26.01	37.08	80.72	49.74	96.80	26.57	77.68	44.99
+ Uniform (OpenWebMath)	26.07	40.09	80.77	50.09	96.70	27.65	78.41	46.62
+ DSIR	25.76	36.63	80.98	48.98	96.50	26.73	72.54	44.27
+ QuRating	25.96	37.84	80.43	50.60	96.80	27.65	77.71	44.97
+ AutoDS	25.84	40.66	80.09	49.74	96.70	27.96	77.19	**47.07**
Mistral-7B Base	29.81	52.39	84.68	57.25	97.40	30.26	83.58	54.30
+ Uniform (OpenWebMath)	29.17	52.17	84.18	56.66	97.20	31.03	83.82	54.91
+ DSIR	29.22	52.62	84.72	57.25	97.30	30.26	73.76	53.54
+ QuRating	28.89	52.01	85.48	57.76	97.30	31.18	82.81	54.03
+ AutoDS	29.06	52.30	84.18	55.20	96.80	31.03	83.12	**55.19**

2024), which itself was curated by simple heuristics and a trained classifier.

2. **DSIR** (Xie et al., 2023b): A KL-divergence-based data selection approach that compares source datasets to a target domain. Here, we use the Pile's Wikipedia split (Gao et al., 2020) as the target to compute domain relevance.

3. **Qurating** (Wettig et al., 2024): A reward-model-based method that ranks candidate training examples by educational value, selecting those with the highest scores.

Training Setup. We use the codebase from LLaMA-Factory (Zheng et al., 2024) alongside DeepSpeed ZeRO-2 Stage (Rajbhandari et al., 2020) to train on nodes with 8 A800 GPUs. The global batch size is set to 256. We conduct smaller-scale experiments with 0.3–0.4B tokens for a preliminary evaluation and larger-scale experiments up to $\sim$ 2.5B tokens for more extensive comparisons. For Mistral-7B pretraining on $\sim$ 300M tokens, we use a cosine learning rate schedule peaking at 5×10^{-6} with a 3% warm-up ratio. For the 2.5B-token experiments, we use a constant learning rate of 1×10^{-6} for Mistral-7B and 1×10^{-5} for Gemma-2B, following recommended practices for continual pretraining.

We evaluate at every 100 updates (about 52M tokens) using a standard evaluation harness (Gao et al., 2023a; Beeching et al., 2023).

4.2 Continual Pretraining Results

Preliminary Evaluation on Mistral-7B. We first conduct a smaller-scale experiment on Mistral-7B-v0.1, continually pretrained with three epochs of either *Uniform (OpenWebMath)* or our top-scoring subset (*AutoMathText*, 328.9M tokens). Figure 6 plots the training loss evolution. We observe that AutoDS data yield faster and more pronounced drops in perplexity. Table 1 reports the zero-shot MATH test accuracy both before and after supervised fine-tuning (SFT) on MetaMathQA (Yu et al., 2024). The AutoDS filter consistently outperforms uniform sampling (e.g., 13.68% $\rightarrow$ 16.14% vs. 10.50% $\rightarrow$ 14.26% on MATH), demonstrating that higher-quality data selection facilitates stronger mathematical reasoning.

Larger-Scale Training (2.5B Tokens). Next, we scale up to $\sim$ 2.5B tokens of math data, comparing four methods: *Uniform (OpenWebMath)*, DSIR, Qurating, and AutoDS. We fine-tune Gemma-2B, LLaMA2-7B, and Mistral-7B for one epoch. Figure 1 visualizes the relative improvements on GSM8K (Cobbe et al., 2021), BBH (Suzgun et al., 2022), and MATH (Hendrycks et al., 2021). Table 2 confirms that our auto-selected data offer consistently stronger performance, particularly on MATH and GSM8K. Notably, on Mistral-7B, AutoDS achieves 16.14% MATH accuracy, surpassing the uniform baseline at 14.26% and demonstrating about 2.36$\times$ higher token efficiency.

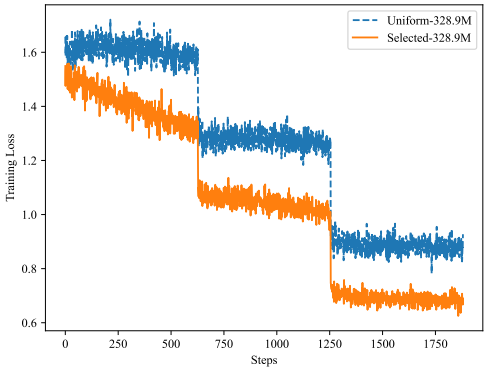

Figure 6: Training-loss evolution when continuing Mistral-7B with either uniform OpenWebMath or AUTODS (328.9M tokens). The loss drops more quickly for our auto-selected data.

4.3 Evaluation on Broader Tasks

We also investigate how improvements in mathematical reasoning might transfer to other cognitive domains such as commonsense and reading comprehension. Table 3 reports results on three representative tasks: CommonsenseQA, ARC-Challenge, and OpenBookQA. While the gains are less pronounced than in math-focused benchmarks, AUTODS tends to either match or slightly surpass the baselines, confirming that focusing on high-quality math data does not degrade general language capabilities.

5 Related Work

Mathematical Language Models and Datasets. Recent advances in chain-of-thought prompting (Radford et al., 2019; Wei et al., 2022; Wang et al., 2023a; Fu et al., 2023; Gao et al., 2023b; Yao et al., 2023; Zhang et al., 2023; Gou et al., 2024) have substantially improved the reasoning capacity of large language models (LLMs). However, most efforts in this line of research focus on eliciting latent reasoning skills through prompting alone rather than augmenting such skills by continuously pretraining on specialized corpora. The creation of high-quality mathematical datasets has played a key role in driving LLMs toward more sophisticated mathematical comprehension and problem-solving abilities. Foundational work in this domain includes the AMPS dataset (Hendrycks et al., 2021), which benchmarks multi-step mathematics questions, and Proof-Pile (Azerbayev et al., 2023), which provides a large-scale corpus of mathematical texts and proofs. Building upon these resources, the Llemma model (Azerbayev et al., 2024) specifically targets continual pretraining on math-oriented data, including OpenWebMath (Paster et al., 2024), to refine complex reasoning skills. Despite such progress, efficiently identifying and leveraging the most instructive mathematical data remains an ongoing challenge.

Data Selection for Language Modeling. Data selection strategies have been explored extensively to improve training efficiency and effectiveness in language modeling. Early efforts by Brown et al. (2020) and Chowdhery et al. (2023) filtered large-scale web data using binary classifiers to favor more reliable or domain-relevant content (e.g., Wikipedia and books). More targeted approaches incorporate domain-specific filtering methods or heuristics: for instance, Minerva (Lewkowycz et al., 2022) applies rules for identifying mathematical text, while DSIR (Xie et al., 2023b) employs importance sampling based on KL divergence to adapt a general corpus to a desired domain. In parallel, DoReMi (Xie et al., 2023a) optimizes domain weights with a proxy model to reduce worst-case excess loss; however, its assumption of relatively high perplexity data may not hold for math or code corpora, whose entropy is inherently lower. Elsewhere, Gunasekar et al. (2023) and Li et al. (2023) used GPT-4 to annotate the educational value of code data and then trained a random forest classifier for data filtering. Qurating (Wettig et al., 2024) proposes a reward-model-based approach to rank training examples automatically. In contrast to these techniques, the present work introduces a fully autonomous data selection framework that relies solely on zero-shot generative classification by base language models, foregoing any reliance on human or model-labeled supervision.

6 Conclusion

We introduced **AutoDS**, an autonomous data selection framework that transforms base language models into zero-shot "generative classifiers" for filtering mathematical texts. By relying solely on model logits to assign real-valued scores, our approach avoids the need for human-annotated labels and enables more fine-grained curation than conventional binary classification methods. Through extensive evaluations, we found that continually pretraining language models on this self-selected corpus markedly enhances mathematical reasoning skills while consuming significantly fewer tokens. Moreover, the improvements on MATH,

GSM8K, and BBH underscore the effectiveness of AutoDS in identifying and prioritizing instructive content. Looking ahead, we plan to extend AutoDS to broader domains to further explore how self-supervised data selection can advance specialized NLP tasks. We hope that making our *AutoMathText* dataset publicly available will foster further research on scalable and autonomous data selection approaches for domain-specific training.

Limitations

Although **AutoDS** effectively curates mathematical texts without human annotations, it depends on the reliability of a large base language model's logits, which may introduce bias when selecting or discarding documents. Furthermore, while we observe improvements on standard math benchmarks, the framework's performance gains may not seamlessly transfer to other specialized domains without careful prompt engineering or domain adaptation.

Ethical Statement

All data used in this work come from publicly accessible sources, and personal or sensitive content is excluded whenever possible. Although our approach may inherit biases from underlying models and data, caution is advised when applying it in high-stakes or real-world settings.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Amittai Axelrod. 2017. Cynical selection of language model training data. *arXiv preprint arXiv:1709.02279*.

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. 2023. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. 2024. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira,

Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

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

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. 2020. Selection via proxy: Efficient data selection for deep learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

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

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. Tora: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Aspen K Hopkins, Alex Renda, and Michael Carbin. 2023. Can llms generate random numbers? evaluating llm sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*.

Edward J. Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. 2024. Amortizing intractable inference in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association Computational Linguistics*. ACL.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances*

in Neural Information Processing Systems, 35:3843–3857.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew Chi-Chih Yao. 2024. Augmenting math word problems via iterative question composing. *arXiv preprint arXiv:2401.09003*.

Miaofeng Liu, Yan Song, Hongbin Zou, and Tong Zhang. 2019. Reinforced training data selection for domain adaptation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1957–1968.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *openai.com*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 372–382. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zengzhi Wang, Rui Xia, and Pengfei Liu. 2023b. Generative ai for math: Part i–mathpile: A billion-token-scale pretraining corpus for math. *arXiv preprint arXiv:2312.17120*.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023b. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao. 2023. Meta prompting for ai systems. *arXiv preprint arXiv:2311.11482*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A More Related Work

Data Selection in Broader Domains. Beyond language modeling, data selection is broadly recognized as an effective method to improve model performance across domains such as vision and domain adaptation. The Moore-Lewis approach (Moore and Lewis, 2010; Axelrod, 2017) pioneered the use of cross-entropy differentials between domain-specific and general-purpose LMs for selective data sampling. Similarly, discrepancies in feature space and n-gram distributions have guided data selection in machine translation and other tasks (Jiang and Zhai, 2007; Liu et al., 2019; Ruder and Plank, 2017). In computer vision, curriculum learning (Bengio et al., 2009) and submodular optimization (Wei et al., 2015) have provided structured ways to curate datasets, while recent prioritized selection methods (Coleman et al., 2020; Mindermann et al., 2022) refine training efficiency by focusing on examples that maximize model improvements. Our proposed method draws inspiration from these broader data selection paradigms but uniquely leverages base LLMs as zero-shot generative classifiers, providing a scalable, domain-specific selection mechanism without human or trained classifier inputs.

Table 4: Comparison of continual pretrained models using different data selection methods on commonsense reasoning tasks.

Model & Selection Method	HellaSwag (10-shot)	PIQA (6-shot)	WinoGrande (15-shot)
Gemma-2B Base	48.30	70.67	60.54
+ Uniform	52.91	76.71	66.38
+ DSIR	52.95	77.15	66.61
+ QuRating	53.10	77.53	66.38
+ AutoDS	52.82	77.42	66.61
LLaMA2-7B Base	58.88	79.43	75.85
+ Uniform	58.43	79.54	75.30
+ DSIR	58.38	78.84	75.37
+ QuRating	58.79	79.00	74.66
+ AutoDS	58.28	78.18	74.51
Mistral-7B Base	62.82	82.10	81.22
+ Uniform	62.21	82.21	80.19
+ DSIR	63.10	81.94	81.37
+ QuRating	62.64	81.99	80.11
+ AutoDS	62.72	82.21	80.03

B More on Experiments

B.1 More Experimental Results

B.2 Prompts

B.3 Alternative Score functions

One can use alternative scoring functions corresponding to different partition functions, such as the formula shown below.

“
<system>
You are ChatGPT, the most capable large language model equipped with extensive expertise in mathematics and coding, particularly skilled in complex reasoning and problem-solving. In the following interaction, I will provide you with a text excerpt from the arXiv website. Your task is to evaluate whether this text contains elements of mathematical intelligence and if it is suitable for educational purposes for YOURSELF in the field of mathematics. Please respond with only YES or NO
</system>

User: {
“Title": “{title}",
“Abstract": “{abstract}",
“Text": “{text}"
}
1. Does the text contain elements of mathematical intelligence? Reply with only YES or NO
2. Is the text suitable for educational purposes for YOURSELF in the field of mathematics? Reply with only YES or NO
”
Assistant: 1.

Figure 7: Prompt for selecting the papers from arXiv.org.

“
<system>
You are ChatGPT, the most capable large language model equipped with extensive expertise in mathematics and coding, particularly skilled in complex reasoning and problem-solving. In the following interaction, I will provide you with a code excerpt from a website. Your task is to evaluate whether this code contains elements of mathematical intelligence and if it is suitable for educational purposes for YOURSELF in the field of mathematics. Please respond with only YES or NO
</system>

User: {
“url": “{url}",
“text": “{text}"
}
1. Does the code contain elements of mathematical intelligence? Reply with only YES or NO
2. Is the code suitable for educational purposes for YOURSELF in the field of mathematics? Reply with only YES or NO
”
Assistant: 1.

Figure 8: Prompt for selecting code snippets from GitHub.

$$\text{LM-Score}_{\text{alternative}}(\cdot) = \frac{\exp(\max(\text{logit}(\text{`YES'}), \text{logit}(\text{`Yes'})))}{\exp(\max(\text{logit}(\text{`YES'}), \text{logit}(\text{`Yes'}))) + \exp(\max(\text{logit}(\text{`NO'}), \text{logit}(\text{`No'})))}.$$

“
Define a function called `isOdd` that takes an argument, $n \in \mathbb{N}$, and returns a proposition that asserts that n is odd. The function will thus be a predicate on values of type $\mathbb{N}$. Hint: a number is odd if it's one more than an even number.

$$\text{def isOdd}(n : \mathbb{N}) : \text{Prop} := \exists m : \text{nat}, 2 \cdot m + 1 = n$$

To test your predicate, use “example” to write and prove `isOdd(15)`.

```
example : isOdd 15 :=
begin
unfold isOdd,
apply exists.intro 7,
apply rfl,
end
```

Define `isSmall` : $\mathbb{N} \to$ Prop, to be a predicate that is true exactly when the argument, n, is such that $n = 0 \vee n = 1 \vee n = 2 \vee n = 3 \vee n = 4 \vee n = 5$. (Don't try to rewrite this proposition as an inequality; just use it as is.)

$$\text{def isSmall}(n : \mathbb{N}) : \text{Prop} := n = 0 \vee n = 1 \vee n = 2 \vee n = 3 \vee n = 4 \vee n = 5$$

”
...
[LM-Score (Q_1, Q_2): 0.963]
“
Define the universes and variables for the context of our category and functor:

universes v u

variables $\{J : \text{Type } v\}$ [small_category J] $\{C : \text{Type } u\}$ [category.$\{v\}$ C] $(F : J \to C)$

Enter noncomputable theory mode and define the initial object's colimit cocone:

```
def is_initial.colimit_cocone {j : J} (hj : is_initial j)
  [has_colimit F] [\forall (a b : J) (f : a \rightarrow b),
  is_iso (F.map f)] :
  cocone F :=
{ X := F.obj j,
  \iota :=
  { app := $\lambda$ i, inv (F.map $ hj.to _),
    naturality' := begin
      intros a b f,
      dsimp,
      simp only [is_iso.eq_inv_comp, is_iso.comp_inv_eq,
      category.comp_id],
      simp_rw ← F.map_comp,
      congr' 1,
      apply hj.hom_ext,
    end } }
```

”
...
[LM-Score (Q_1, Q_2): 0.439]

Figure 9: Examples contain Lean4 code. It is difficult for human beings without math expertise to judge the educational value of these examples for language models on learning mathematics.

C Appendix for Examples

C.1 Web Subset

C.2 Code Subset

Example:
"# In mathematics the monomial basis of a polynomial ring is its basis (as vector space or free module over the field or ring of coefficients) that consists in the set of all monomials. The monomials form a basis because every polynomial may be uniquely written as a finite linear combination of monomials (this is an immediate consequence of the definition of a polynomial). One indeterminate The polynomial ring K[x] of the univariate polynomial over a field K is a K-vector space, which has $1, x, x^2, x^3, \ldots$ as an (infinite) basis. More generally, if K is a ring, K[x] is a free module, which has the same basis. The polynomials of degree at most d form also a vector space (or a free module in the case of a ring of coefficients), which has $1, x, x^2, \ldots$ as a basis The canonical form of a polynomial is its expression on this basis: $a_0 + a_1x + a_2x^2 + \ldots + a_dx^d$, or, using the shorter sigma notation: $\sum_{i=0}^{d} a_ix^i$. The monomial basis in naturally totally ordered, either by increasing degrees $1 < x < x^2 < \cdots$, or by decreasing degrees $1 > x > x^2 > \cdots$. Several indeterminates In the case of several indeterminates $x_1, \ldots, x_n$, a monomial is a product $x_1^{d_1}x_2^{d_2}\cdots x_n^{d_n}$, where the d_i are non-negative integers. Note that, as $x_i^0 = 1$, an exponent equal to zero means that the corresponding indeterminate does not appear in the monomial; in particular $1 = x_1^0x_2^0\cdots x_n^0$ is a monomial. ..."
LM-Score (Q_1): 0.987, **LM-Score (Q_2):** 0.662, **LM-Score (Q_1, Q_2):** 0.653

Example: Commutative Property Of Addition
"Commutative Property Of Addition 2. If A is an n×m matrix and O is a m×k zero-matrix, then we have: AO = O Note that AO is the n×k zero-matrix. Matrix Matrix Multiplication 11:09. We have 1. To understand the properties of transpose matrix, we will take two matrices A and B which have equal order. The identity matrix is a square matrix that has 1's along the main diagonal and 0's for all other entries. In a triangular matrix, the determinant is equal to the product of the diagonal elements. This matrix is often written simply as I, and is special in that it acts like 1 in matrix multiplication. Is the Inverse Property of Matrix Addition similar to the Inverse Property of Addition? The identity matrices (which are the square matrices whose entries are zero outside of the main diagonal and 1 on the main diagonal) are identity elements of the matrix product. Learning Objectives. In fact, this tutorial uses the Inverse Property of Addition and shows how it can be expanded to include matrices! Keywords: matrix; matrices; inverse; additive; additive inverse; opposite; Background Tutorials. ..."
LM-Score (Q_1): 0.991, **LM-Score (Q_2):** 0.954, **LM-Score (Q_1, Q_2):** 0.946

Example: Comparing the magnitudes of expressions
"# Comparing the magnitudes of expressions of surds I recently tackled some questions on maths-challenge / maths-aptitude papers where the task was to order various expressions made up of surds (without a calculator, obviously). I found myself wondering whether I was relying too much on knowing the numerical value of some common surds, when a more robust method was available (and would work in more difficult cases). For example, one question asked which is the largest of: (a) $\sqrt{10}$ (b) $\sqrt{2}+\sqrt{3}$ (c) $5-\sqrt{3}$ In this case, I relied on my knowledge that $\sqrt{10} \approx 3.16$ and $\sqrt{2} \approx 1.41$ and $\sqrt{3} \approx 1.73$ to find (a) ≈ 3.16, (b) $\approx$ 3.14 and (c) $\approx$ 3.27 so that the required answer is (c). But this seemed inelegant: I felt there might be some way to manipulate the surd expressions to make the ordering more explicit. I can't see what that might be, however (squaring all the expressions didn't really help). ..."
LM-Score (Q_1): 0.991, **LM-Score (Q_2):** 0.946, **LM-Score (Q_1, Q_2):** 0.937

Example: In Calculus, function derivatives
"# In Calculus, how can a function have several different, yet equal, derivatives? I've been pondering this question all night as I work through some problems, and after a very thorough search, I haven't found anything completely related to my question. I guess i'm also curious how some derivatives are simplified as well, because in some cases I just can't see the breakdown. Here is an example: $f(x) = \dfrac{x^2-6x+12}{x-4}$ is the function I was differentiating. Here is what I got: $f'(x) = \dfrac{x^2-8x+12}{(x-4)^2}$ which checks using desmos graphing utility. Now, when I checked my textbook(and Symbolab) they got: $f'(x) = 1 - \dfrac{4}{(x-4)^2}$ which also checks on desmos. To me, these derivatives look nothing alike, so how can they both be the equal to the derivative of the original function? Both methods used the quotient rule, yet yield very different results. Is one of these "better" than the other? I know that it is easier to find critical numbers with a more simplified derivative, but IMO the derivative I found seems easier to set equal to zero than the derivative found in my book.I also wasn't able to figure out how the second derivative was simplified, so I stuck with mine. I'm obviously new to Calculus and i'm trying to understand the nuances of derivatives. ..."
LM-Score (Q_1): 0.985, **LM-Score (Q_2):** 0.950, **LM-Score (Q_1, Q_2):** 0.936

Example: Math help on cubics

"# Math Help - working backwards - cubics 1. ## working backwards - cubics Write an equation that has the following roots: 2, -1, 5 Answer key: $x^3 - 6x^2 + 3x + 10 = 0$ For quadratic equations, I use the sum and product of roots, this is a cubic equation, how do I solve this? Thanks. 2. Originally Posted by shenton Write an equation that has the following roots: 2, -1, 5 Answer key: $x^3 - 6x^2 + 3x + 10 = 0$ For quadratic equations, I use the sum and product of roots, this is a cubic equation, how do I solve this? Thanks. $(x - 2)(x + 1)(x - 5)$ 3. Thanks! That turns out to be not as difficult as imagined. I thought I needed to use sum and products of roots to write the equation, it does makes me wonder a bit why or when I need to use sum and products of roots. 4. Write an equation that has the following roots: 2, -1, 5 Is there any other way to solve this other than the (x-2)(x+1)(x-5) method? If we have these roots: 1, $1 + \sqrt{2}$, $1 - \sqrt{2}$ the (x - 1) $(x - 1 - \sqrt{2})$ $(x - 1 + \sqrt{2})$ method seems a bit lenghty. When we expand (x - 1) $(x - 1 - \sqrt{2})$ $(x - 1 + \sqrt{2})$ the first 2 factors, it becomes: $(x^2 - x - x\sqrt{2} - x + 1 + \sqrt{2})$ $(x - 1 + \sqrt{2})$ collect like terms: $(x^2 - 2x - x\sqrt{2} + 1 + \sqrt{2})$ $(x - 1 + \sqrt{2})$ To further expand this will be lenghty, my gut feel is that mathematicians do not want to do this - it is time consuming and prone to error. There must be a way to write an equation other than the above method. Is there a method to write an equation with 3 given roots (other than the above method)? ..."

LM-Score (Q_1): 0.991, **LM-Score (Q_2):** 0.943, **LM-Score (Q_1, Q_2):** 0.935

Example: Work and time

"# Work and time, when work is split into parts I'm stuck on a particular type of work and time problems. For example, 1) A,B,C can complete a work separately in 24,36 and 48 days. They started working together but C left after 4 days of start and A left 3 days before completion of the work. In how many days will the work be completed? A simpler version of the same type of problem is as follows: 2) A can do a piece of work in 14 days while B can do it in 21 days. They begin working together but 3 days before the completion of the work, A leaves off. The total number of days to complete the work is? My attempt at problem 2: A's 1 day work=1/14 and B's 1 day work= 1/21 Assume that it takes 'd' days to complete the entire work when both A and B are working together. Then, (1/14 + 1/21)*d= 1 -> d=42/5 days. But it is stated that 3 days before the completion of the work, A left. Therefore, work done by both in (d-3) days is: (1/14 + 1/21)*(42/5 - 3)= 9/14 Remaining work= 1- 9/14 = 5/14 which is to be done by B alone. Hence the time taken by B to do (5/14) of the work is: (5/14)*21 = 7.5 days. Total time taken to complete the work = (d-3) + 7.5 = 12.9 days. However, this answer does not concur with the one that is provided. My Understanding of problem 1: Problem 1 is an extended version of problem 2. But since i think i'm doing problem 2 wrong, following the same method on problem 1 will also result in a wrong answer. Where did i go wrong? ..."

LM-Score (Q_1): 0.991, **LM-Score (Q_2):** 0.941, **LM-Score (Q_1, Q_2):** 0.932

Example: Inequality Involving Sums

"Inequality involving sums with binomial coefficient I am trying to show upper- and lower-bounds on $\frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}\min(i, n-i)$ (where $n \geq 1$) in order to show that it basically grows as $\Theta(n)$. The upper-bound is easy to get since $\min(i, n-i) \leq i$ for $i \in \{0, \dots n\}$ so that $\frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}\min(i, n-i) \leq \frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}i = \frac{n}{2}$. Thanks to Desmos, I managed to find a lower bound, but I am struggling to actually prove it. Indeed, I can see that the function $f(n) = \frac{n-1}{3}$ does provide a lower-bound. One can in fact rewrite $\frac{n-1}{3} = \frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}\frac{2i-1}{3}$. I was thus hoping to show that for each term we have $\frac{2i-1}{3} \leq \min(i, n-i)$, but this is only true if $i \leq \frac{3n+1}{5}$ and not generally for $i \leq n$. I imagine there is a clever trick to use at some point but for some reason, I am stuck here. Any help would be appreciated, thank you! EDIT: Thank you everyone for all the great and diverse answers! I flagged River Li's answer as the "accepted" one because of its simplicity due to the use of Cauchy-Schwartz inequality, which does not require a further use of Stirling's approximation. ..."

LM-Score (Q_1): 0.988, **LM-Score (Q_2):** 0.941, **LM-Score (Q_1, Q_2):** 0.931

Example: Algebraic Manipulation

"# Algebraic Manipulation ## Definition Algebraic manipulation involves rearranging variables to make an algebraic expression better suit your needs. During this rearrangement, the value of the expression does not change. ## Technique Algebraic expressions aren't always given in their most convenient forms. This is where algebraic manipulation comes in. For example: ### What value of x satisfies $5x + 8 = -2x + 43$ We can rearrange this equation for x by putting the terms with x on one side and the constant terms on the other.

$$\begin{aligned} 5x + 8 &= -2x + 43 \\ 5x - (-2x) &= 43 - 8 \\ 7x &= 35 \\ x &= \frac{35}{7} \\ x &= 5 \quad \square \end{aligned}$$

Algebraic manipulation is also used to simplify complicated-looking expressions by factoring and using identities. Let's walk through an example: ### $\frac{x^3+y^3}{x^2-y^2} - \frac{x^2+y^2}{x-y}$. It's possible to solve for x and y and plug those values into this expression, but the algebra would be very messy. Instead, we can rearrange the problem by using the factoring formula identities for $x^3 + y^3$ and $x^2 - y^2$ and then simplifying.

$$\begin{aligned} \frac{x^3 + y^3}{x^2 - y^2} - \frac{x^2 + y^2}{x - y} &= \frac{(x + y)(x^2 - xy + y^2)}{(x - y)(x + y)} - \frac{x^2 + y^2}{x - y} \\ &= \frac{x^2 - xy + y^2 - (x^2 + y^2)}{x - y} \\ &= \frac{-xy}{x - y} \end{aligned}$$

Plugging in the values for xy and $x - y$ gives us the answer of 3. ..."
LM-Score (Q_1): 0.990, **LM-Score (Q_2):** 0.940, **LM-Score (Q_1, Q_2):** 0.931

Example: Finding the minimum number

"# Finding the minimum number of students There are p committees in a class (where $p \geq 5$), each consisting of q members (where $q \geq 6$).No two committees are allowed to have more than 1 student in common. What is the minimum and maximum number of students possible? It is easy to see that the maximum number of student is pq,however I am not sure how to find the minimum number of students.Any ideas? 1) $pq - \binom{q}{2}$ 2) $pq - \binom{p}{2}$ 3) $(p-1)(q-1)$ - Something is missing. Is every student supposed to be on a committee? – JavaMan Aug 31 '11 at 16:24 @DJC:Not mentioned in the question,I guess we may have to consider that to get a solution. – Quixotic Aug 31 '11 at 16:28 @DJC: For the minimum number of students this does not matter. – TMM Aug 31 '11 at 16:30 @Thijs Laarhoven:Yes you are right but as the problem also asked for maximum number I have considered it in my solution. – Quixotic Aug 31 '11 at 16:31 @Thijs, FoolForMath, I guess my question is, should the minimum answer be in terms of p and q? – JavaMan Aug 31 '11 at 16:31 For $1 \leq i \leq p$, let C_i be the set of students on the ith committee. Then by inclusion-exclusion, or more accurately Boole's inequalities, we have

$$\sum_i |C_i| - \sum_{i<j} |C_i C_j| \leq |C_1 \cup C_2 \cup \cdots \cup C_p| \leq \sum_i |C_i|.$$

From the constraints of the problem, this means

$$pq - \binom{p}{2} \leq \#\text{ students} \leq pq.$$

- What is j here?and I can't relate this with your answer. j is also a generic index that runs from 1 to p. The inequalities are also known as Bonferroni inequalities (planetmath.org/encyclopedia/BonferroniInequalities.html), and can apply to cardinalities instead of probabilities. – Byron Schmuland Sep 1 '11 at 14:10 I think the following theorem might be relevant: Theorem. Let $\mathcal{F}$ be a family of subsets of $\{1, \ldots, n\}$ with the property that $|A \cap B| = 1$ for all $A, B \in \mathcal{F}$. Then $|\mathcal{F}| \leq n$. Also this theorem could be relevant as well. - For the case in which $p \leq q + 1$ an arrangement that yields the minimum number of students can be described as follows. Let $P = \{\langle m, n\rangle : 1 \leq m \leq p, 1 \leq n \leq q + 1\}$, and let $S = \{\langle m, n\rangle \in P : m < n\}$. If P is thought of as a $p \times (q + 1)$ grid, ..."
LM-Score (Q_1): 0.985, **LM-Score (Q_2):** 0.863, **LM-Score (Q_1, Q_2):** 0.850

Example: Applied Linear Algebra
"Let $w_1 = (0, 1, 1)$. Expand $\{w_1\}$ to a basis of R^3. I am reading the book, Applied Linear Algebra and Matrix Analysis. When I was doing the exercise of Section3.5 Exercise 7, I was puzzled at some of it. Here is the problem description: Let $w_1 = (0, 1, 1)$. Expand $\{w_1\}$ to a basis of R^3. I don't understand its description well. I think it wants to get a span set like $\{(0, 1, 1), (1, 0, 0), (0, 0, 1)\}$ which is a basis of R^3. And I check the reference answer, which is as followings: $(0, 1, 1), (1, 0, 0), (0, 1, 0)$ is one choice among many. I think what I have done is what question wants. So can anyone tell me am I right or wrong? Thanks sincerely. • I think you are right Apr 16, 2019 at 6:02 There is a kind of 'procedure' for dealing with questions of this kind, namely to consider the spanning set $\{w_1, e_1, e_2, e_3\}$. Consider each vector from left to right. If one of these vectors is in the span of the previous one/s, then throw it out. If not, keep it. So in this case, we start by keeping w_1. Moving to the next vector, e_1 is not in the span of w_1, so we keep it as well. Moving to the next, e_2 is not in the span of the previous two vectors so we keep it as well. Now, considering the vector e_3 we see that it is in fact in the span of the previous three vectors, since $e_3 = w_1 - e_2$. So we throw out the vector e_3 and end up with the basis $\{w_1, e_1, e_2\}$. This explains the solution in the reference answer. Your solution is also correct, however. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ has independent rows. Hence you have found 3 independent vectors in $\mathbb{R}^3$, that is it spans $\mathbb{R}^3$ and it forms a basis. You are correct. $(0, 1, 1), (1, 0, 0), (0, 0, 1)$ is a basis of $\mathbb{R}^3$. Any element (a, b, c) in $\mathbb{R}^3$ can be expressed as $a(1, 0, 0) + b(0, 1, 1) + (c - b)(0, 0, 1)$. If your basis is w_1, w_2, w_3, the textbook's choice is $w_1, w_2, w_1 - w_3$ ..."
LM-Score (Q_1): 0.964, **LM-Score (Q_2):** 0.882, **LM-Score (Q_1, Q_2):** 0.850

Example: Mathematical Analysis
"## Solution to Principles of Mathematical Analysis Chapter 7 Part A ### Chapter 7 Sequences and Series of Functions #### Exercise 1 (By analambanomenos) Let $\{f_n\}$ be a uniformly convergent sequence of bounded functions on a set E. For each n, there is a number M_n such that $|f_n(x)| < M_n$ for all $x \in E$. By Theorem 7.8, there is an integer N such that $|f_n(x) - f_N(x)| < 1$ if $n \geq N$ for all $x \in E$. Let

$$M = \max\{M_1, \ldots, M_N\}.$$

Then for $n \leq N$ and $x \in E$ we have $|f_n(x)| < M + 1$, and for $n \geq N$ and $x \in E$ we have

$$|f_n(x)| \leq |f_n(x) - f_N(x)| + |f_N(x)| < M + 1.$$

That is, the f_n are uniformly bounded by $M + 1$ in E. ..."
LM-Score (Q_1): 0.976, **LM-Score (Q_2):** 0.820, **LM-Score (Q_1, Q_2):** 0.800

Example: Vector equations
"# Vector equations, possible to solve for x? #### Jonsson Hello there, In scalar algebra, I find solving for variables a useful tool. Say ohms law, I want to find R so:

$$U = RI \iff R = \frac{U}{I}$$

Can I do something analogous in vector equations? I.e. May I solve for $\vec{\omega}$ in equations using cross or dot products?

$$\vec{v} = \vec{\omega} \times \vec{r} \iff \vec{\omega} = ?$$

or:

$$\vec{\alpha} \cdot \vec{\beta} = \gamma \iff \vec{\beta} = ?$$

It would be fantastic if I could solve for vectors in some way. Hope you are able to help. Kind regards, Marius #### maajdl Gold Member Solving v=wxr makes sense, since this can be seen as solving 3 equations with 3 unknowns (each components). You can find the solution easily by "multiplying" both sides by r: rxv = rx(wxr) = w (r.r) - r (w.r). ..."
LM-Score (Q_1): 0.950, **LM-Score (Q_2):** 0.842, **LM-Score (Q_1, Q_2):** 0.800

Example: Linear programming

"# If then Constraint 2 Hello all: I want to implement the following constraint in my linear programing model: If A=B then C=1 Else C=0 I have been looking around and there are similar problems but nobody has been helpful to address the 'non equal to' condition. Thank you in advance. asked 27 Sep '14, 17:45 Chicago 33 5 accept rate: 0% 3 As I understand the question, you want c to be binary, and $c = 1$ if and only if $A = B$. I will make a couple of assumptions: There is a (large) positive M such that $|A - B| \leq M$ for every feasible (A, B). There is a (small) positive ϵ such that whenever $A \neq B$, we can assume there is a solution satisfying $|A - B| \geq \epsilon$. Here's the formulation:

$$\begin{aligned} A &\leq B + My - \epsilon z \\ B &\leq A + Mz - \epsilon y \\ c + y + z &= 1 \\ c, y, z &\in \{0, 1\} \end{aligned}$$

Now, if $c = 1$, then $y = z = 0$. In this case, the constraints reduce to $A \leq B$ and $B \leq A$, so $A = B$. Otherwise, $c = 0$. Then $y + z = 1$. There are two cases. ..."

LM-Score (Q_1): 0.950, **LM-Score (Q_2):** 0.842, **LM-Score (Q_1, Q_2):** 0.800

Example: Distance formula

"The distance formula is a formula that is used to find the distance between two points. These points can be in any dimension. The x-z plane is vertical and shaded pink ... If observation i in X or observation j in Y contains NaN values, the function pdist2 returns NaN for the pairwise distance between i and j.Therefore, D1(1,1), D1(1,2), and D1(1,3) are NaN values.. Contents. Print the the distance between two points on the surface of earth: —– Input the latitude of coordinate 1: 25 Input the longitude of coordinate 1: 35 Input the latitude of coordinate 2: 35.5 Input the longitude of coordinate 2: 25.5 The distance between those points is: 1480.08 Flowchart: C++ Code Editor: Contribute your code and comments through Disqus. Interactive Distance Formula applet. Distance Formula Calculator. Find the square root of that sum: $\sqrt{90} = 9.49$. In a 3 dimensional plane, the distance between points (X 1, Y 1, Z 1) and (X 2, Y 2, Z 2) are given. The distance between two points on the three dimensions of the xyz-plane can be calculated using the distance formula The distance formula is derived from the Pythagorean theorem. and: Line passing through two points. Parameters first Iterator pointing to the initial element. Distance between 2 points in 3D space calculator uses Distance between 2 points=$\sqrt{(x2-x1)^2+(y2-y1)^2+(z2-z1)^2}$ to calculate the Distance between 2 points, ..."

LM-Score (Q_1): 0.950, **LM-Score (Q_2):** 0.737, **LM-Score (Q_1, Q_2):** 0.700

Example: Estimate from below of the sine

"# Estimate from below of the sine (and from above of cosine) I'm trying to do the following exercise with no success. I'm asked to prove that $\sin(x) \geq x - \frac{x^3}{2}, \quad \forall x \in [0, 1]$ By using Taylor's expansion, it's basically immediate that one has the better estimate $\sin(x) \geq x - \frac{x^3}{6}, \quad \forall x \in [0, 1]$ as the tail converges absolutely, and one can check that the difference of consecutive terms is positive. I suppose then, there is a more elementary way to get the first one. Question is: how? Relatedly, the same exercise asks me to prove that $\cos(x) \leq \frac{1}{\sqrt{1+x^2}}, \quad \forall x \in [0, 1]$ which again I can prove by using differentiation techniques. But these haven't been explained at that point of the text, so I wonder how to do it "elementary". I showed by comparison of areas that for first quadrant angles $\sin\theta\cos\theta \leq \theta \leq \tan\theta$ If one multiplies the left of these inequalities by 2 it becomes $\sin 2\theta < 2\theta$ so we arrive at $\sin\theta \leq \theta \leq \tan\theta$ Rearrange the right of these inequalities to $\frac{\sin\theta}{\theta} \geq \cos\theta$ or $1 - \frac{\sin\theta}{\theta} \leq 1 - \cos\theta = 2\sin^2\frac{\theta}{2} \leq 2\left(\frac{\theta}{2}\right)^2 = \frac{\theta^2}{2}$ Where we have used the left of the above inequalities above. This rearranges to $\sin\theta \geq \theta - \frac{\theta^3}{2}$ for first quadrant angles. ..."

LM-Score (Q_1): 0.950, **LM-Score (Q_2):** 0.737, **LM-Score (Q_1, Q_2):** 0.700

Example: Force on side of pool from water

"Force on side of pool from water Given a pool with dimensions $\ell \times w \times h$, I am trying to derive an equation that will yield the force by the water on the sides of the pool, namely $\ell \times h$ or $w \times h$. For the side of the pool with dimensions $\ell \times h$, I started by using the familiar equation for pressure $F = PA$. Plugging in the expression for hydrostatic pressure for P gives $F = \rho ghA = \rho gh(\ell \times h) = \boxed{\rho g \ell h^2}$. Is my reasoning, and corresponding solution correct? Hydrostatic pressure changes with height. You have just multiplied by area, which means that you have assumed it to be constant. Instead, you should integrate over the area. You'll get an extra 1/2 term for the force. – Goobs Sep 15 '15 at 4:21 As @Goobs says, the pressure force is 0 at the top of the water line and increases to $\rho\ g\ y\ dA$ on a surface of area dA at depth y. Since this pressure increases linearly from 0 to $\rho\ g\ y$ the average force on the wall is the average of the start and end: so, it is half of this value, and the total pressure is $\frac{1}{2}\rho gh(h\ell)$. Would this be correct? $\int dF = \int_0^H \rho g A\ dh = \rho g \ell \int_0^H h\ dh = \boxed{\frac{1}{2}\rho g H^2}$ – rgarci0959 Sep 15 '15 at 4:51 Yes. For bonus points you would write it as $\int dA\ \rho\ g\ h$ to start with, as that's one of those forces that you "know" is correct ..."

LM-Score (Q_1): 0.987, **LM-Score (Q_2):** 0.662, **LM-Score (Q_1, Q_2):** 0.653

Example: Lagrange's Interpolation Method

```
X = [0, 20, 40, 60, 80, 100]
Y = [26.0, 48.6, 61.6, 71.2, 74.8, 75.2]
n = len(X)-1
# Degree of polynomial = number of points - 1
print("X =", X)
print("Y =", Y, end='\n\n')
xp = float(input("Find Y for X = "))
# For degree of polynomial 3, number of points n+1 = 4:
# L[1] = (x-x2)/(x1-x2) * (x-x3)/(x1-x3) * (x-x4)/(x1-x4)
# L[2] = (x-x1)/(x2-x1) * (x-x3)/(x2-x3) * (x-x4)/(x2-x4)
# L[3] = (x-x1)/(x3-x1) * (x-x2)/(x3-x2) * (x-x4)/(x3-x4)
# L[4] = (x-x1)/(x4-x1) * (x-x2)/(x4-x2) * (x-x3)/(x4-x3)
# L[i] *= (x-xj)/(xi-xj) where i, j = 1 to n+1 and j != i
# y += Y[i]*L[i] where i = 1 to n+1
# List index 0 to n
# ~~~~~~~~~~~~~~~~~~~~~~~~ Method 1: Using for loop ~~~~~~~~~~~~~~~~~~~~~~~~
yp = 0
# Initial summation value
for i in range(n+1):
    L = 1
    # Initial product value
    for j in range(n+1):
        if j == i:
            continue
        # j == i gives ZeroDivisionError
        L *= (xp - X[j]) / (X[i] - X[j]) yp += Y[i]*L
# ~~~~~~~~~~~~~~~~~~~~ Method 2: Using numpy array, prod ~~~~~~~~~~~~~~~~~~~~
from numpy import array, prod
X = array(X, float)
Y = array(Y, float)
yp = 0
for Xi, Yi in zip(X, Y):
    yp += Yi * prod((xp - X[X != Xi]) / (Xi - X[X != Xi]))
```

LM-Score (Q_1): 0.977, **LM-Score (Q_2):** 0.959, **LM-Score (Q_1, Q_2):** 0.937

Example: Scientific Computing Theory

```
# Question 01, Lab 04
# AB Satyaprakash - 180123062
# imports ----------------------------------------------------------------
from sympy.abc import x
from sympy import cos, exp, pi, evalf, simplify
# functions --------------------------------------------------------------
def midpointRule(f, a, b):
    return ((b-a)*f.subs(x, (b-a)/2)).evalf()

def trapezoidalRule(f, a, b):
    return (((b-a)/2)*(f.subs(x, a)+f.subs(x, b))).evalf()

def simpsonRule(f, a, b):
    return (((b-a)/6)*(f.subs(x, a)+4*f.subs(x, (a+b)/2)+f.subs(x, b))).evalf()

# program body
# part (a) I = integrate cosx/(1+cos^2x) from 0 to pi/2 -- exact value = 0.623225
f = cos(x)/(1 + cos(x)**2)
a, b = 0, pi/2
print('To integrate {} from {} to {}'.format(simplify(f), a, b))
print('Evaluated value of integral using Midpoint rule is', midpointRule(f, a, b))
print('Evaluated value of integral using Trapezoidal rule is', trapezoidalRule(f, a, b))
print('Evaluated value of integral using Simpson rule is', simpsonRule(f, a, b))
print('Exact value = 0.623225\n')

# part (b) I = integrate 1/(5+4cosx) from 0 to pi -- exact value = 1.047198
f = 1/(5 + 4*cos(x))
a, b = 0, pi
print('To integrate {} from {} to {}'.format(simplify(f), a, b))
print('Evaluated value of integral using Midpoint rule is', midpointRule(f, a, b))
print('Evaluated value of integral using Trapezoidal rule is', trapezoidalRule(f, a, b))
print('Evaluated value of integral using Simpson rule is', simpsonRule(f, a, b))
print('Exact value = 1.047198\n')

# part (c) I = integrate exp(-x^2) from 0 to 1 -- exact value = 0.746824
f = exp(-x**2)
a, b = 0, 1
```

LM-Score (Q_1): 0.982, **LM-Score (Q_2):** 0.946, **LM-Score (Q_1, Q_2):** 0.929

Example: Fourth Order Runge-Kutta (RK4) Method

```
from numpy import exp, linspace, empty
f = lambda x: exp(x-2) - 3 # Analytical Solution
dy = lambda x, y: y+3 # Equation to be solved, y' = y+3
x = 2 # Lower limit, [2
xn = 4 # Upper limit, 4]
y = -2 # Initial condition, y(2) = -2
h = 0.1 # Width of each division, step size
n = int((xn-x)/h) # Number of divisions of the domain
# Plot Arrays
xp = linspace(x, xn, n+1)
# Divides from x to xn into n+1 points
yp = empty(n+1, float)
yp[0] = y
print('x \t\ty(RK4) \t\ty(Analytical)')
# Header of Output
print('%f \t% f \t% f' % (x, y, f(x)))
# Initial x and y
for i in range(1, n+1):
    K1 = h * dy(x,y)
    K2 = h * dy(x + h/2, y + K1/2)
    K3 = h * dy(x + h/2, y + K2/2)
    K4 = h * dy(x + h, y + K3)
    y += 1/6*(K1 + 2*K2 + 2*K3 + K4) # y(x+h) = y(x) + 1/6(K1+2K2+2K3+K4)
    yp[i] = y
    x += h # x for next step,
    x = x + h
    print('%f \t% f \t% f' % (x, y, f(x)))
    # ~~~~~~~~~~~~~~~~~~~~~~~~~ Plotting the function ~~~~~~~~~~~~~~~~~~~~~~~~~
    import matplotlib.pyplot as plt # pyplot.
    plt.plot(xp, yp, 'ro', xp, f(xp)) # Default plot is continuous blue line
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend(['RK4', 'Analytical'])
    plt.show()
```

LM-Score (Q_1): 0.982, **LM-Score (Q_2):** 0.945, **LM-Score (Q_1, Q_2):** 0.928

Example: Real roots of the quadratic equation

```
from math import sqrt
from numpy.testing import assert_equal, assert_allclose
def real_quadratic_roots(a, b, c):
"""
Find the real roots of the quadratic equation a x^2 + b x + c = 0, if they exist.
Parameters ----------
a : float Coefficient of x^2
b : float Coefficient of x^1
c : float Coefficient of x^0
Returns -------
roots : tuple or float or None The root(s) (two if a genuine quadratic, one if linear, None otherwise)
Raises ------
NotImplementedError If the equation has trivial a and b coefficients, so isn't solvable.
"""
    discriminant = b**2 - 4.0*a*c
    if discriminant < 0.0:
        return None
    if a == 0:
        if b == 0:
            raise NotImplementedError("Cannot solve quadratic with both a" " and b coefficients equal to
  0.")
        else: return -c / b

    x_plus = (-b + sqrt(discriminant)) / (2.0*a)
    x_minus = (-b - sqrt(discriminant)) / (2.0*a)
    return x_plus, x_minus

def test_no_roots():
"""
Test that the roots of x^2 + 1 = 0 are not real.
"""
roots = None
assert_equal(real_quadratic_roots(1, 0, 1), roots, err_msg="Testing x^2+1=0; no real roots.")
```

LM-Score (Q_1): 0.977, **LM-Score (Q_2):** 0.950, **LM-Score (Q_1, Q_2):** 0.928