STEM-PoM: Evaluating Language Models Math-Symbol Reasoning in Document Parsing

Jiaru Zou[1]†, **Qing Wang**[1], **Pratyush Thakur**[1], **Nickvash Kani**[1]†
[1] University of Illinois Urbana-Champaign
{jiaruz2, kani}@illinois.edu

Abstract

Advances in large language models (LLMs) have spurred research into enhancing their reasoning capabilities, particularly in math-rich STEM (Science, Technology, Engineering, and Mathematics) documents. While LLMs can generate equations or solve math-related queries, their ability to fully understand and interpret abstract mathematical symbols in long, math-rich documents remains limited. In this paper, we introduce STEM-PoM, a comprehensive benchmark dataset designed to evaluate LLMs' reasoning abilities on math symbols within contextual scientific text. The dataset, sourced from real-world ArXiv documents, contains over 2K math symbols classified as main attributes of variables, constants, operators, and unit descriptors, with additional sub-attributes including scalar/vector/matrix for variables and local/global/discipline-specific labels for both constants and operators. Our extensive experiments demonstrate that state-of-the-art LLMs achieve an average accuracy of 20–60% under in-context learning and 50–60% with fine-tuning, highlighting a substantial gap in their ability to classify mathematical symbols. By improving LLMs' mathematical symbol classification, STEM-PoM further enhances models' downstream mathematical reasoning capabilities. The code and data are available at https://github.com/jiaruzouu/STEM-PoM.

1 Introduction

Large language models (LLMs) have demonstrated exceptional reasoning abilities across numerous fields (Huang and Chang, 2022; Hadi et al., 2023, 2024; Li et al., 2024; Zheng et al., 2024; He et al., 2024b). With the increasing shift towards applying LLMs to complex tasks (Brown, 2020; Kojima et al., 2022; Sui et al., 2023; Zou et al., 2024; He et al., 2024c), the need for supplementary data beyond the general pre-trained datasets has become increasingly important. Among these, mathematical reasoning tasks (English, 2013; Ilany et al., 2010) have recently drawn the attention of several researchers (Imani et al., 2023; Ahn et al., 2024; Zhang et al., 2024; Lu et al., 2023). In particular, Part-of-Math Tagging (Youssef, 2017), the mathematical analog to part-of-speech tagging (Schmid, 1994), where mathematical tokens are classified according to a given taxonomy of attributes, continues to gain interest with the integration of LLMs math reasoning.

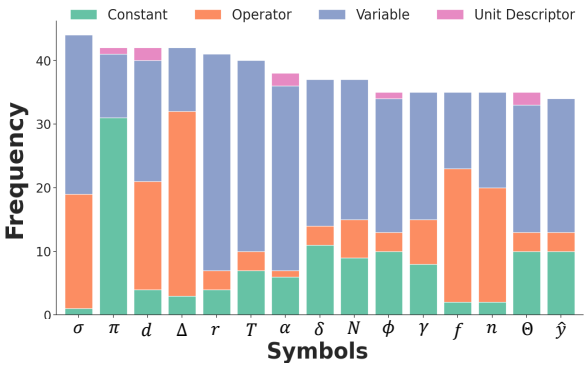

Figure 1: Total frequency (appearances) of the top-15 mathematical symbols in 1,000 randomly sampled ArXiv math-rich documents. Each math symbol contemporary belongs to multiple main attribute categories depending on its related context and mathematical expression. This illustrates the **contextual polymorphism** of a single math symbol.

However, despite this growing interest, Part-of-Math Tagging currently still lacks the foundational datasets that are crucial for supporting advanced NLP tasks (Youssef, 2017; Shan and Youssef, 2021, 2024). In addition, integrating mathematical language into NLP models remains a substantial challenge (Alshamari and Youssef, 2020; Meadows and Freitas, 2022), especially in the realm of document parsing (Dridan and Oepen, 2013; Lam et al., 2008; Zhang et al., 2019). Traditional semantic parsing methods such as LaTeXML (Miller, 2011)

†Corresponding authors

Findings of the Association for Computational Linguistics: ACL 2025, pages 8183–8199
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

often fall short when applied to math-rich documents, where precision and structured syntax are paramount (Hamel et al., 2022; Paster et al., 2023; Wang et al., 2023). These methods struggle to accurately perform pattern matching between abstract mathematical symbols and their corresponding XML tag notations.

Similarly, recent advanced LLMs, such as ChatGPT (Liu et al., 2023), also face difficulties in understanding and reasoning with abstract mathematical symbols due to their contextual polymorphism (Ditchfield, 1994; Fiore and Hamana, 2013; Murase et al., 2023), as demonstrated in Figure 1. As an intuitive example, in the linear equation: $y = mx + p$, y is categorized as a variable. Whereas in the cross-entropy loss function: $\mathcal{L}(x, y) = -\sum_{i=1}^{N} y_i \log(x_i)$, the symbol y represents the fixed target labels, which is considered a constant for a given dataset. Without the corresponding contextual information of a mathematical symbol, LLMs are unable to distinguish between different attributes of the symbol and cannot effectively process related mathematical reasoning tasks. Thus, tagging math symbols within domain-specific contexts is essential for language models.

In this paper, we introduce a novel benchmark dataset, **STEM-PoM**, designed to evaluate the reasoning capabilities of language models on mathematical symbols across different domains. The STEM-PoM dataset consists of 2,109 mathematical token instances extracted from a random sampling of 10,000 arXiv manuscripts, which are math-rich documents spanning domains such as Mathematics, Physics, Chemistry, and more. For each dataset instance, we provide a specific mathematical symbol with the corresponding symbol order in the document, main and sub-level attributes, and the related text information from the original ArXiv paper. Each mathematical symbol in the dataset is classified according to two levels of attributes (Wikipedia, 2023). The first-level attribute categorizes the symbol as variable, constant, operator, or unit descriptor. The second-level attribute further classifies the symbol into one of six types based on its first-level category: scalar, vector, matrix, local, global, or discipline-specific. Figure 2 illustrates the category distribution of the dataset. To further enrich the STEM-PoM dataset with additional arXiv manuscripts and other math-rich document resources, we also design the **STEM-PoM Labeler**, a feasible toolkit to assist dataset generation by automatically searching, extracting, and recording hand-labeled mathematical symbols with their corresponding context from STEM articles.

We conduct extensive experiments on the STEM-PoM dataset to assess the mathematical reasoning abilities of a large amount of open- and closed-source vanilla language models. Our experimental results indicate that existing language models generally struggle with understanding mathematical symbols, even when provided with relevant context. Extending the context length to a certain degree and fine-tuning on symbol-related material can enhance their ability to classify mathematical symbols. We further examine the relationship between the model's mathematical symbol classification and mathematical reasoning capabilities by evaluating its performance on challenging downstream tasks, such as OlympiadBench (He et al., 2024a), after enhancing its classification accuracy through fine-tuning on STEM-PoM. Last, we provide detailed dataset analysis and case studies to investigate the influence factors such as context length on the ability of language models to understand mathematical symbols. In summary, our **main contributions** are:

- We propose a novel task combining Part-of-Math Tagging with document parsing to assess LLMs' mathematical symbol classification abilities.
- We introduce STEM-PoM, a benchmark dataset comprised of over 2,000 annotated instances extracted from math-rich documents spanning fields including Mathematics, Physics, and Chemistry. Each math symbol in the dataset is meticulously labeled with multiple hierarchical attributes, reflecting its role and context within the original document.
- We evaluate LLMs' mathematical reasoning on STEM-PoM, revealing performance variations across different models and highlighting the role of math symbol understanding in enhancing LLMs' mathematical reasoning abilities.

2 Backgrounds

Part-of-Math (PoM) Tagging. The part-of-math tagging task draws inspiration from similar tagging tasks such as part-of-speech tagging (Schmid, 1994). In the PoM context, the goal is to label individual mathematical tokens or expressions in math formulas with their corresponding mathematical roles. Youssef (2017) collected mathematical content, such as formula representation and

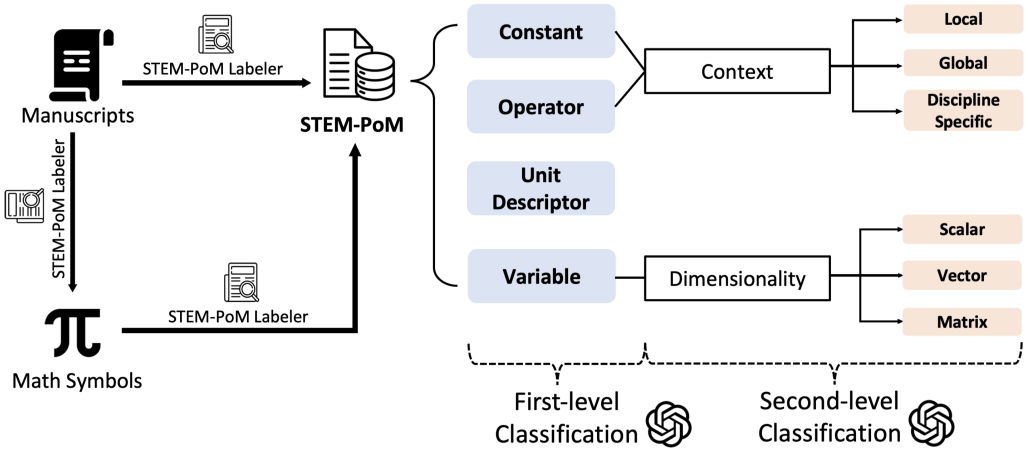

Figure 2: The overall pipeline for constructing the STEM-POM dataset. We extract math symbols with corresponding text information to formulate the dataset. Each math symbol is initially classified into one of four primary categories based on its definition. Then, the symbol is further categorized into secondary categories by the context in which it appears or by the symbol's dimensionality. An LLM is evaluated via the first-level and second-level classification tasks.

tagging for specific mathematical formula translations and verifications, including converting formulae into semantic LaTeX or testing with tools like CAS (Computer Algebra Systems). However, this focus on structured and narrow formula translations does not align with the broader, more diverse text-based tasks required to assess NLP models, due to the lack of scalability features in the collected math symbols. Shan and Youssef (2021, 2024) recently evaluated the potential of leveraging LLMs for automated annotation and Part-of-Math tagging of math symbols conducted on the Digital Library of Mathematical Functions (DLMF) (Lozier, 2003). Since the source of math symbols is only one manuscript, the mathematical tokens collected only have a single classification type and are self-consistent. In contrast, our dataset incorporates the inherent messiness of published literature across several STEM subjects, where these domain-specific math symbols can have multiple classifications or meanings depending on the discipline and related contextual information.

Math Symbol Annotation. In the context of mathematical symbol annotations, a common approach is to simply extract definitions for mathematical tokens, transforming part-of-math tagging into a name-entity-recognition (NER) task (Asakura et al., 2021; Shan and Youssef, 2024). While NER does evaluate an AI model's ability to process text, it is fundamentally different from part-of-math tagging, which requires the model to understand and classify mathematical tokens based on a specific taxonomy. Unlike NER, part-of-math tagging tasks defined in our dataset demand deeper comprehension of the mathematical symbol.

Large Language Models. Pre-trained large language models (LLMs) have become a cornerstone in modern NLP (Rosenfeld, 2000; Zhao et al., 2023). Early approaches were based on N-gram models, but with the advent of distributed word embeddings (Bengio et al., 2000; Mikolov, 2013), neural language models gained prominence. The scalability and performance improvements introduced by these models and the availability of vast textual data have enabled the unsupervised pre-training of LLMs. These models, often referred to as foundation models (Radford et al., 2019; Kojima et al., 2022), can then be fine-tuned on smaller, task-specific datasets to adapt them for various downstream applications. For STEM-POM, we apply one traditional sequence-based NLP model, LSTM (Graves and Graves, 2012), and several most updated LLMs for our dataset evaluation.

3 STEM-POM Dataset

In this section, we introduce our constructed benchmark dataset, STEM-POM. First, we describe the source data used for extracting mathematical symbols and text information in Section 3.1. Next, we outline the data annotation process in Section 3.2 and present the dataset statistics in Section 3.2. Finally, we provide details on our dataset labeling tool STEM-PoM labeler in Section 3.4.

Statistic	Number
Total Symbols	**2,109**
Unit Descriptor	**129**
Constant	**384**
- Local	171
- Global	121
- Discipline Specific	92
Operator	**363**
- Local	181
- Global	105
- Discipline Specific	77
Variable	**1,233**
- Scalar	601
- Vector	599
- Matrix	33
Avg symbols per article	4.7
Avg tokens per sentence	31.8
Avg tokens per math symbol	1.07

Table 1: STEM-POM Dataset Statistics

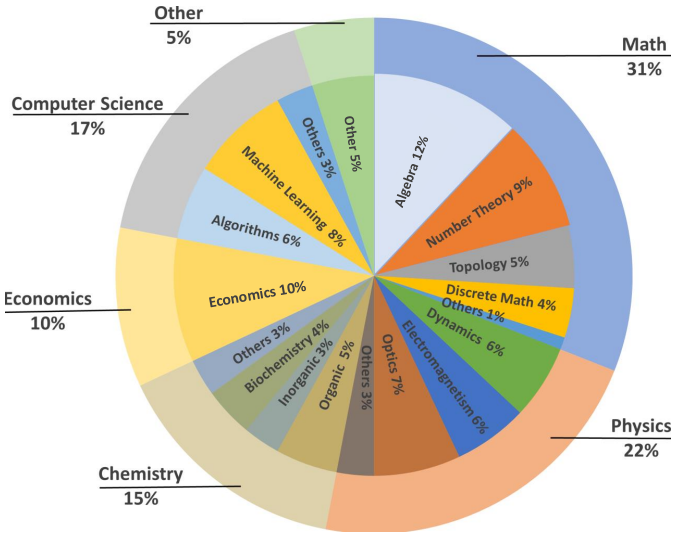

Figure 3: Discipline Distribution from Source ArXiv

File Name	Symbol Order	Symbol	Main Attribute	Sub Attribute	Related Contents
9509/adap-org9509001.html	0	f	Constant	Global	...1/f noise was discovered...
9509/adap-org9509001.html	1	Δ	Operator	Global	...can be quantified by studying the displacement ΔX
9509/adap-org9509001.html	2	X	Unit Descriptor	-	...can be quantified by studying the displacement ΔX
9509/adap-org9509001.html	3	t	Variable	Scalar	..after t steps, we can...
...	...	...	...	...	...

Table 2: STEM-POM Dataset Structure

3.1 Source Dataset

The first crucial step in constructing the dataset is selecting high-quality mathematical symbols. For STEM-POM, we primarily collect these symbols from two sources: 1. *Public math-symbol datasets*, where we directly utilize candidate mathematical symbols from the mathematical token definition extraction benchmark, MTDE (Hamel et al., 2022). 2. *Raw ArXiv papers* (Clement et al., 2019), where we identify and extract mathematical symbols with corresponding context including math equations, math symbol definitions, and related text sentences from the dataset. The detailed description of each raw dataset is provided below:

MTDE (Hamel et al., 2022) contains approximately 10,000 entries of mathematical symbol names along with their defined contexts. Each entry includes a 'short' definition and a 'long' definition. A short definition is a single-word definition, while a long definition consists of one or more words. The data was collected through random sampling from mathematical and scientific arXiv preprint manuscripts, covering a broad range of disciplines such as Physics, Computer Science, and Biology. Note that all math symbols in the MTDE dataset are pre-filtered and well-defined by the authors, ensuring our annotation process minimizes the risk of misinformation or repetition issues. In our dataset pre-processing, we also make sure the candidate data is generated via a corpus crawler and subsequently pruned and cleaned manually.

ArXiv Paper Dataset (Kohlhase et al., 2024) contains 1.7 million arXiv articles, spanning a wide range of disciplines, including Mathematics, Physics, Chemistry, Economics, and Computer Science. We will provide a detailed explanation of how STEM-POM is constructed from the raw source dataset in the next subsection.

3.2 Dataset Construction

To construct our dataset, we randomly sampled 10,000 articles from the ArXiv Paper Dataset across various domains. We manually verified that each manuscript is math-rich, containing numerous mathematical expressions and symbols. Using pre-filtered mathematical symbols from the MTDE, we engaged human-expert annotators to utilize the STEM-PoM labeler to: (1) align each mathematical

symbol with its source paper, (2) identify relevant contextual information for each symbol, and (3) address disambiguation and aliasing by annotating all possible meanings for symbols with multiple interpretations and specifying the context in which each meaning applies. Through the rigorous selection process, 453 of the 10,000 articles are identified as the final source articles with high-quality contextual matches for the symbols. Each article contains an average of 4.7 annotated mathematical symbols (2109 in total).

After obtaining the mathematical symbols, we categorize each symbol into different attributes and assign surrounding context information to construct the STEM-PoM dataset. Specifically, we first extract the file name and symbol order for each mathematical symbol. Then, for each symbol, we extract the contexts in which the symbol appears, using several predefined lengths. Following this, we manually classify each symbol into four main attribute (first-level) categories: Variable, Constant, Operator, and Unit Descriptor. For the Variable, Constant, and Operator, we further categorize them into sub-attribute (second-level) categories. A variable is classified as a Vector, Scalar, or Matrix, while a Constant or Operator is categorized as Local, Global, or Discipline-Specific. Table 2 outlines the overall dataset structure. We manually examine each entry of the dataset thoroughly to ensure its robustness and correctness. The overall dataset pipeline is demonstrated in Figure 2. We also provide a detailed explanation of the dataset structure in Appendix A.1 and the definitions of each level's attributes in Appendix A.2.

Dataset Statistics We summarize the key statistics of our dataset in this section. Table 1 presents the categorical statistics, including the math symbols along with their first- and second-level attributes. The distribution of Variables, Constants, Operators, and Unit Descriptors is 58.5%, 18.2%, 17.2%, and 6.1%, respectively. In addition, Figure 3 illustrates the discipline distribution of the source arXiv papers. Our dataset covers mathematical symbols from various fields, including Mathematics, Physics, Chemistry, Economics, Computer Science, etc.

3.3 Dataset Quality Control

During the dataset construction, we manually evaluate the quality and applicability of the annotated data. Specifically, we process the evaluation process through both consistency checks and Inter-annotator agreements.

Consistency Checks. To ensure reliability, we perform the following consistency checks: each article is assigned to domain-specific experts for labeling, and we provide precise definitions and examples for each label and category to guide the process. After labeling, we randomly select a subset of the labeled data for consistency evaluation. Additionally, we analyze data points with inconsistent labels, facilitate discussions among annotators to resolve ambiguities and reach a consensus, and refine the annotation guidelines to address any common sources of confusion.

Inter-annotator Agreements. Regarding the IAA in our human annotation process, we engage 33 domain experts as our annotators to annotate domain-specific articles. After the initial labeling process, the rest annotators (5 Annotator/Discipline on average) from the same domain reviewed the annotated data by switching their labeling components. For the Inter-annotator Agreement, we measure with Cohen's Kappa value and ensure the value ranges from 0.81 to 1 (0.903 on average across all labeled data), regarding the dataset's mathematical symbols and their corresponding attributes. To maintain accuracy, annotators have access to the original papers through the STEM-PoM labeler and annotate the attributes of the data based on the content of the original papers. This thorough review process reinforces the reliability of our annotated dataset.

3.4 STEM-PoM Labeler

We developed a toolkit named STEM-PoM Labeler to assist human experts in retrieving math symbols along with their corresponding article contexts, annotating the dataset, and conducting inter-annotator consistency checks to ensure agreement in dataset labeling. For a detailed description of the toolkit's design, please refer to Appendix A.3.

4 Experiments

4.1 Setups

Models. To thoroughly evaluate our dataset across models with varying parameter sizes, we utilize the following models: LSTM (Graves, 2013), Mixtral-8x7B-v0.1 (Jiang et al., 2024), Llama2-13B (Touvron et al., 2023), Llama3.1-70B (Dubey et al., 2024), Claude-3.5-Sonnet (Anthropic, 2024),

Models	Context Length	Overall	Variable	Constant	Operator	Unit Descriptor
LSTM (Graves, 2013)	One Sentence	18.7%	24.5%	13.2%	10.3%	27.1%
	Ten Sentences	22.6%	28.1%	16.8%	15.5%	30.2%
	Full Manuscript	-	-	-	-	-
Llama2-13B (Touvron et al., 2023)	One Sentence	36.8%	24.1%	39.3%	41.4%	42.7%
	Ten Sentences	42.7%	35.6%	39.8%	46.9%	48.5%
	Full Manuscript	45.9%	38.2%	42.8%	50.1%	52.4%
Mistral-8x7B (Jiang et al., 2024)	One Sentence	47.3%	38.5%	41.7%	52.9%	56.2%
	Ten Sentences	49.8%	41.8%	45.9%	58.6%	56.7%
	Full Manuscript	53.6%	45.7%	48.9%	61.4%	58.2%
Llama3.1-70B (Dubey et al., 2024)	One Sentence	48.9%	41.3%	44.6%	48.5%	61.5%
	Ten Sentences	53.0%	44.8%	48.8%	54.7%	63.7%
	Full Manuscript	51.7%	42.7%	43.2%	55.2%	65.8%
Claude3.5-Sonnet (Anthropic, 2024)	One Sentence	63.7%	58.6%	62.5%	65.7%	67.8%
	Ten Sentences	65.9%	61.3%	64.3%	67.9%	70.2%
	Full Manuscript	66.7%	62.9%	65.8%	68.6%	69.3%
GPT-3.5-turbo (Achiam et al., 2023)	One Sentence	56.8%	51.5%	53.5%	59.4%	62.4%
	Ten Sentences	58.7%	54.5%	53.6%	61.3%	65.1%
	Full Manuscript	60.6%	57.2%	56.6%	63.2%	65.2%
GPT-4o (Hurst et al., 2024)	One Sentence	64.9%	60.5%	64.2%	64.9%	70.1%
	Ten Sentences	67.4%	63.7%	66.1%	66.4%	73.5%
	Full Manuscript	68.5%	64.2%	67.8%	68.1%	73.8%

Table 3: First-level classification accuracy with various context lengths. Here. **One sentence/Ten Sentences/Full Manuscript** refers to the context size of completed sentences as contextual information for each math symbol.

Models	Variable			Constant			Operator		
	Scalar	Vector	Matrix	Local	DS	Global	Local	DS	Global
LSTM (Graves, 2013)	13.8%	15.1%	17.2%	19.2%	17.8%	22.2%	16.6%	11.3%	14.6%
Llama2-13B (Touvron et al., 2023)	27.3%	24.4%	21.8%	33.6%	31.5%	33.6%	32.4%	28.3%	32.7%
Mistral-8x7B (Jiang et al., 2024)	36.9%	35.8%	21.6%	34.8%	31.2%	37.8%	36.4%	34.8%	35.7%
Llama3.1-70B (Dubey et al., 2024)	38.2%	34.1%	26.7%	37.6%	35.2%	36.1%	39.1%	32.3%	40.2%
Claude3.5-Sonnet (Anthropic, 2024)	53.2%	49.7%	55.8%	55.9%	53.1%	49.6%	56.3%	52.2%	55.9%
GPT-3.5-turbo (Achiam et al., 2023)	44.5%	45.8%	48.3%	48.5%	42.9%	44.3%	48.4%	43.5%	49.7%
GPT-4o (Hurst et al., 2024)	54.6%	51.3%	58.6%	58.4%	54.1%	56.2%	60.5%	57.3%	58.5%

Table 4: Second-level classification accuracy with full manuscript input (Ten-sentence input for LSTM). We abbreviate "Discipline Specific" as "DS".

GPT-3.5-Turbo-0125[1] (Achiam et al., 2023), and GPT-4o-2024-08-06[2] (Hurst et al., 2024).

Evaluation Metrics. We apply the *Precision Accuracy* as our metric for the mathematical symbol classification task, the metric can be formulated as:

$$\textit{Precision Accuracy} = \frac{\textit{Number of correct predictions}}{\textit{Total number of samples}}$$

Training & Inference Details. We evaluate several models under both pre-training and fine-tuning settings. Specifically, we train an LSTM model with varying layers and apply the LoRA method, a parameter-efficient fine-tuning (PEFT) technique, to several updated LLMs. We also evaluate additional LLMs under the in-context learning setting. The training and model parameter details are provided in Appendix B.

[1] https://platform.openai.com/docs/models/gpt-3-5-turbo

[2] https://platform.openai.com/docs/models/gpt-4o

4.2 First-Level Classification Results

Table 3 summarizes the accuracy results of various models across different context lengths. To extract context information, we implement a surrounding window provided for domain human expert annotators to meticulously select the most rel-

Models (fine-tuned)	Context Length	Overall	Variable	Constant	Operator	Unit Descriptor
Llama-2-13B (Touvron et al., 2023)	Ten Sentences	45.5%	38.4%	41.7%	50.6%	51.3%
Mixtral-8x7B (Jiang et al., 2024)	Ten Sentences	55.7%	46.3%	51.9%	64.2%	60.3%
Llama3.1-70B (Dubey et al., 2024)	Ten Sentences	62.4%	56.6%	54.5%	70.8%	67.5%
GPT-3.5-turbo (Achiam et al., 2023)	Ten Sentences	66.9%	65.4%	66.6%	71.3%	64.5%
GPT-4o (Hurst et al., 2024)	Ten Sentences	70.3%	71.3%	74.1%	75.2%	68.1%

Table 5: First-level classification accuracy with various context lengths. Here. "One sentence/Ten Sentences/Full Manuscript" refers to the context size of completed sentences used as contextual information for testing each mathematical symbol.

evant completed sentences surrounding the math symbol. Through the careful context selection process, we ensure that the information included in the prompts is accurate and contextually relevant. The result shows that the small-parameter-size language model such as the LSTM struggles with lower accuracy, achieving between 18.7% and 22.6%. In contrast, larger models, such as Claude3.5-Sonnet and GPT-4o, show marked improvements as context length increases, with accuracy consistently above 63.7% and up to 73.8%. We also found that the performance gap between models remains consistent as context length increases. To demonstrate, GPT-4o outperforms Llama3.1-70B by 16.0%, 14.4%, and 16.8% for context lengths of one sentence, ten sentences, and the full manuscript, respectively. This consistent performance gap suggests that larger models with more pre-trained knowledge, such as GPT-4o, exhibit superior scalability with longer contexts.

Another notable observation is that the overall performance gain from increasing context length is more pronounced in smaller models, such as Llama2-13B and Mistral-8x7B, which have less pre-trained knowledge. These models benefit more from extended context as they rely on additional information to compensate for their limited pre-training. Larger models like GPT-4o and Claude3.5-Sonnet, which come with extensive pre-trained knowledge, show relatively smaller performance gains as context length increases.

4.3 Second-Level Classification Results.

Table 4 shows second-level classification accuracy with full manuscript input. In this experiment setting, we assume that the model got the first-level classification correct. By horizontally comparing the same model performance on different sub-attribute classifications, we find that the attribute Constants are generally easier to classify compared to Variables and Operators across all sizes of models, as seen by the overall higher accuracy in Constant-related tasks. However, Matrix and DS classification continue to present challenges, even for the largest models, indicating that certain structures, as well as content types within manuscripts, remain difficult to categorize accurately at the sub-attribute level.

Overall, performance across all models on both first-level and second-level classification tasks shows a clear trend of improvement with increasing context length, highlighting the importance of context for accurately classifying mathematical symbols. Additionally, both small and large-size language models show a relatively higher accuracy in identifying Unit Descriptors and Operators compared to Variables and Constants, indicating that symbols with more distinct contextual or syntactical patterns are easier for models to classify. Through the above results, we aim to gain insights into the extent to which different category attributes of mathematical symbols influence LLMs' understanding of math-rich documents by correctly classifying the symbols in real-world scenarios.

4.4 Fine-tuning on STEM-POM

We fine-tune four different sizes of LLMs on STEM-POM, as detailed in Table 5. Comparing these results with those in Table 3, we observe that additional training improves model classification accuracy for mathematical symbols when provided with more context information. However, continuously adding contextual knowledge to the training samples does not always lead to performance gains. We analyze this relationship further in Section 4.6.

4.5 Downstream Math Reasoning

After evaluating the performance of STEM-POM across different LLMs, we further investigate its effectiveness in enhancing LLMs' math reasoning

capabilities. Specifically, we aim to address the following questions:

- **Q1:** What is the relationship between an LLM's ability to classify mathematical tokens and its real-world mathematical reasoning skills?
- **Q2:** Does improving an LLM's mathematical token classification capability enhance corresponding math reasoning abilities?

Models	GSM8K	MATH	OlympiadBench	**Avg.**
Llama2-13B	42.5%	29.1%	11.5%	27.7%
+ LoRA (STEM-PoM)	44.6% (↑ 2.1)	31.3% (↑ 2.2)	13.4% (↑ 1.9)	29.8% (↑ 2.1)
Mixtral-8x7B	72.4%	32.6%	13.7%	39.6%
+ LoRA (STEM-PoM)	74.1% (↑ 1.7)	34.1% (↑ 1.5)	16.4% (↑ 2.7)	41.5% (↑ 1.9)
Llama3.1-70B	91.6%	47.1%	26.4%	55.0%
+ LoRA (STEM-PoM)	93.2% (↑ 1.6)	48.8% (↑ 1.7)	28.2% (↑ 1.8)	56.7% (↑ 1.7)
GPT-4o	94.3%	88.7%	39.6%	74.2%
+ LoRA (STEM-PoM)	95.2% (↑ 0.9)	88.9% (↑ 0.2)	41.2% (↑ 1.6)	75.1% (↑ 0.9)

Table 6: Evaluation of LLMs on downstream mathematical reasoning tasks. **+ LoRA (STEM-PoM)** indicates that the corresponding model is first fine-tuned using LoRA on STEM-PoM before being evaluated on downstream mathematical reasoning tasks. The improvements from LoRA fine-tuning on STEM-PoM are highlighted in darkgreen (↑ x).

To investigate these questions, we select challenging mathematical problems from GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024a) as our downstream tasks. We evaluate the model's performance before and after LoRA fine-tuning on STEM-PoM. The evaluation is conducted using greedy decoding without tool integration. The primary metric reported is pass@1 accuracy with 3-shots CoT prompting. Full implementation details are shown in Appendix C.

Table 6 presents the experimental results on three mathematical reasoning datasets. A comparison across different models reveals that LLMs demonstrate superior performance on STEP-PoM and also generally achieve higher accuracy in mathematical reasoning tasks. This suggests that **higher math token classification accuracy generally correlates with better mathematical reasoning performance.** Additionally, all models exhibit performance gains after being fine-tuned on STEM-PoM. This suggests that **enhancing LLMs' understanding of mathematical symbols contributes to improved reasoning abilities in mathematical problem-solving.** The experimental results align with the objective of our constructed dataset, which aims to fundamentally enhance LLMs' mathematical reasoning through Part-of-Math Tagging.

4.6 Analysis on STEM-PoM

Model size(layers)	Variable	Constant	Operator	Unit Descriptor
128	24.5%	13.2%	10.3%	27.1%
256	28.7%	17.9%	15.7%	32.5%
512	34.2%	23.2%	24.9%	40.0%
1024	46.5%	35.9%	44.2%	51.3%

Table 7: LSTM first-level classification accuracy based on different model sizes

Model Performance vs Model Size. Table 7 presents the classification accuracy of an LSTM model for first-level classification across different model sizes, ranging from 128 to 1024 layers. Note that we set the input context length to be one sentence. The results show a clear positive correlation between the model size and classification accuracy across all four categories. For the smallest model (128 layers), the accuracy ranges from 10.3% for the Operator class to 27.1% for the Unit Descriptor class. As the model size increases, the performance improves notably, with the largest model (1024 layers) achieving a relatively high-performance gain in accuracy, ranging from 35.9% for the Constant class to 51.3% for the Unit Descriptor class. The most substantial improvements are observed in the Operator category, where accuracy increases from 10.3% for 128 layers to 44.2% for 1024 layers. These results suggest that larger model sizes are more effective in capturing complex patterns. We further analyze the impact of input sequence length on model performance and compare the effectiveness of fine-tuning versus in-context learning on STEM-PoM. Please refer to D for full analysis.

Case Study and Further Discussion. Please refer to Appendix E and F for the additional case study and discussions on STEM-PoM.

5 Conclusion

In this paper, we introduce STEM-PoM, a comprehensive benchmark for evaluating language models' mathematical reasoning abilities to classify math symbols from scientific texts. The dataset includes over 2,000 math instances sourced from ArXiv papers. Extensive experiments show that the best-performing model, achieves only 73.8% and 60.5% for first and second-level classification accuracy, highlighting the challenge of extracting and categorizing math symbols from large text corpora.

6 Limitations

While we introduce a new benchmark dataset to evaluate LLMs' reasoning abilities with math symbols, several challenges identified in our experimental evaluations warrant further investigation. First, as discussed in Section 4.3, even the most advanced LLMs struggle to accurately recognize complex math symbol attributes, such as matrices and discipline-specific constants or operators. To address this limitation, additional prompting or encoding methods, such as (Fu et al., 2023), need to be explored. Second, as evidenced by Table 3 and 4, when computational resources are constrained, leveraging smaller language models to process extensive mathematical context while maintaining performance levels comparable to larger models like GPT-4o presents a significant challenge. Through our benchmark, STEM-POM, we intend to investigate these directions in future work to enhance LLMs' mathematical reasoning capabilities within the scope of Part-of-Math Tagging.

7 Ethics Statement

The dataset proposed in this paper is derived from publicly available datasets that were released under licenses permitting reuse for research purposes. In constructing our dataset, we have taken care to ensure that all data sources comply with applicable legal and ethical standards, including privacy and data protection regulations. We have only used datasets that are free from personally identifiable information (PII) and sensitive attributes and have not modified or added any data that could compromise the anonymity or privacy of individuals. Furthermore, we encourage responsible use of the dataset and have made it available solely for academic and research purposes, with the expectation that future users will adhere to similar ethical guidelines.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Fatimah Alshamari and Abdou Youssef. 2020. Astudy into math document classification using deep learning.

Anthropic. 2024. Claude 3.5 sonnet. Accessed: 2024-10-14.

Takuto Asakura, Yusuke Miyao, Akiko Aizawa, and Michael Kohlhase. 2021. Miogatto: A math identifier-oriented grounding annotation tool. In *CICM Workshops*.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Colin B Clement, Matthew Bierbaum, Kevin P O'Keeffe, and Alexander A Alemi. 2019. On the use of arxiv as a dataset. *arXiv preprint arXiv:1905.00075*.

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*.

Glen Jeffrey Ditchfield. 1994. Contextual polymorphism.

Rebecca Dridan and Stephan Oepen. 2013. Document parsing: Towards realistic syntactic analysis. In *Proceedings of The 13th International Conference on Parsing Technologies (IWPT 2013)*, pages 127–133.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Lyn D English. 2013. *Mathematical reasoning: Analogies, metaphors, and images*. Routledge.

Marcelo Fiore and Makoto Hamana. 2013. Multiversal polymorphic algebraic theories: syntax, semantics, translations, and equational logic. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 520–529. IEEE.

Yingnan Fu, Tingting Liu, Ming Gao, and Aoying Zhou. 2023. Edsl: An encoder-decoder architecture with symbol-level features for printed mathematical expression recognition. In *International Conference on Document Analysis and Recognition*, pages 134–151. Springer.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

André Greiner-Petter. 2023. *Making Presentation Math Computable: A Context-Sensitive Approach for Translating LaTeX to Computer Algebra Systems*. Springer Nature.

André Greiner-Petter, Moritz Schubotz, Corinna Breitinger, Philipp Scharpf, Akiko Aizawa, and Bela Gipp. 2022. Do the math: Making mathematics in wikipedia computable. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4384–4395.

Muhammad Usman Hadi, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, et al. 2024. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Emma Hamel, Hongbo Zheng, and Nickvash Kani. 2022. An evaluation of nlp methods to extract mathematical token descriptors. In *International Conference on Intelligent Computer Mathematics*, pages 329–343. Springer.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 2024a. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Xinrui He, Yikun Ban, Jiaru Zou, Tianxin Wei, Curtiss B Cook, and Jingrui He. 2024b. Llm-forest: Ensemble learning of llms with graph-augmented prompts for data imputation. *arXiv preprint arXiv:2410.21520*.

Xinyi He, Jiaru Zou, Yun Lin, Mengyu Zhou, Shi Han, Zejian Yuan, and Dongmei Zhang. 2024c. Cocost: Automatic complex code generation with online searching and correctness testing. *arXiv preprint arXiv:2403.13583*.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S Weld, and Marti A Hearst. 2021. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

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. *arXiv preprint arXiv:2103.03874*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Bat-Sheva Ilany, Bruria Margolin, et al. 2010. Language and mathematics: Bridging between natural language and mathematical language in solving problems in mathematics. *Creative Education*, 1(03):138.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Michael Kohlhase et al. 2024. arxmliv project. https://kwarc.info/projects/arXMLiv/. Accessed: 2024-09-17.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Tak Cheung Lam, Jianxun Jason Ding, and Jyh-Charn Liu. 2008. Xml document parsing: Operational and performance characteristics. *Computer*, 41(9):30–37.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017.

Daniel W Lozier. 2003. Nist digital library of mathematical functions. *Annals of Mathematics and Artificial Intelligence*, 38:105–119.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Jordan Meadows and André Freitas. 2022. A survey in mathematical language processing. *arXiv preprint arXiv:2205.15231*.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

B Miller. 2011. Latexml the manual. *Web document*.

Yuito Murase, Yuichi Nishiwaki, and Atsushi Igarashi. 2023. Contextual modal type theory with polymorphic contexts. In *European Symposium on Programming*, pages 281–308. Springer Nature Switzerland Cham.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*.

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.

Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Helmut Schmid. 1994. Part-of-speech tagging with neural networks. *arXiv preprint cmp-lg/9410018*.

Ruocheng Shan and Abdou Youssef. 2021. Towards math terms disambiguation using machine learning. In *Intelligent Computer Mathematics: 14th International Conference, CICM 2021, Timisoara, Romania, July 26–31, 2021, Proceedings 14*, pages 90–106. Springer.

Ruocheng Shan and Abdou Youssef. 2024. Using large language models to automate annotation and part-of-math tagging of math equations. In *International Conference on Intelligent Computer Mathematics*, pages 3–20. Springer.

Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. 2023. Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning. *arXiv preprint arXiv:2312.09039*.

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*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.

Wikipedia. 2023. Glossary of mathematical symbols. Accessed: 2024-09-17.

Abdou Youssef. 2017. Part-of-math tagging and applications. In *International Conference on Intelligent Computer Mathematics*, pages 356–374. Springer.

Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2019. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2287–2305.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, et al. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? *arXiv preprint arXiv:2403.14624*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lecheng Zheng, Baoyu Jing, Zihao Li, Hanghang Tong, and Jingrui He. 2024. Heterogeneous contrastive learning for foundation models and beyond. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6666–6676.

Jiaru Zou, Mengyu Zhou, Tao Li, Shi Han, and Dongmei Zhang. 2024. Promptintern: Saving inference costs by internalizing recurrent prompt during large language model fine-tuning. *arXiv preprint arXiv:2407.02211*.

A STEM-POM Dataset Supplementary Materials

A.1 Dataset Definitions in Table 2

File Name: This attribute serves as a reference point, indicating the source of the file. Specifically, it denotes the arXiv article from which the dataset extracts its contents.
Symbol Order: This component records the sequence in which mathematical symbols appear within the article. By capturing the ordinal position of each symbol, we facilitate a structured analysis of the symbols' progression and their contextual relationships within the document.
Symbols: This field encapsulates the mathematical symbols themselves, predominantly consisting of Greek letters, albeit inclusive of additional characters. The precise documentation of these symbols is paramount for the subsequent analytical processes.
Main and Sub Attributes: These attributes categorize each mathematical symbol into specific classes, delineating a hierarchical structure within the dataset. This classification scheme is vital for understanding the symbols' roles and relationships within the mathematical discourse.
Related Contents: This segment comprises the words or sentences surrounding each symbol, embodying a critical resource for our model training. The contextual information surrounding each symbol is indispensable, as it imbues our models with a deeper understanding of each symbol's application and significance within the mathematical narrative.

A.2 First-Level and Second-Level Attributes Definition

Constant: A value that does not change in a mathematical expression. **Local Constant:** Constant that is specific to a particular system or model, such as the gravitational constant in a simulation of a specific planetary system. **Global Constant:** Constant that is applicable in all contexts, like the speed of light in a vacuum. **Discipline-specified Constant:** Constant that applies to particular fields of study, for instance, Planck's constant in quantum mechanics. **Operator:** A symbol that operates on one or more operands. **Local Operators:** Operator that is applied in a localized or specific context within a discipline, like a self-defined operation in matrix processing. **Global Operators:** Operators that is used broadly across different disciplines, like the addition or multiplication operator. **Discipline-specified Operators:** Operator that is unique to certain fields, such as the Hamiltonian operator in quantum physics. **Variable:** A symbol that represents an unknown or changeable quantity in a mathematical expression. **Scalar:** A quantity that has only magnitude, no direction. **Vector:** A quantity that has both magnitude and direction. **Matrix:** A rectangular array of numbers or symbols arranged in rows and columns.

A.3 STEM-PoM Labeler

During the dataset construction, a pivotal step involves the meticulous annotation of each mathematical symbol with corresponding tags. This process, inherently labor-intensive and repetitive, necessitates a systematic approach to mitigate the workload and facilitate collaboration among the research team members. To address these challenges, we developed a labeling pipeline designed to streamline the dataset construction process. The UI design is shown in Figure 4. The functionalities are delineated below:

File Reading. We initiate the data importing operation progress by importing files from the designated arXiv folder, ensuring a structured and accessible repository of mathematical documents for subsequent processing.

Symbol Identification and Contextualization. For each file, we enumerate and display essential information: the current file being processed, the total number of symbols within, the sequence number of the current symbol, the graphical representation of the symbol, and the contextual content surrounding the symbol. This feature aids in providing a comprehensive overview and facilitates accurate symbol annotation.

Annotation Interface. We then present a user-friendly interface offering a set of predefined tagging options for each symbol. Through the designed interface, we easily select the most appropriate tag from these options, standardizing the labeling process and enhancing the consistency of the dataset.

Data Recording. Upon the selection of a tag for a symbol, We record this association by appending a new line to the dataset, capturing the symbol along with its assigned tag. This systematic data recording ensures the integrity and scalability of the MTCE dataset.

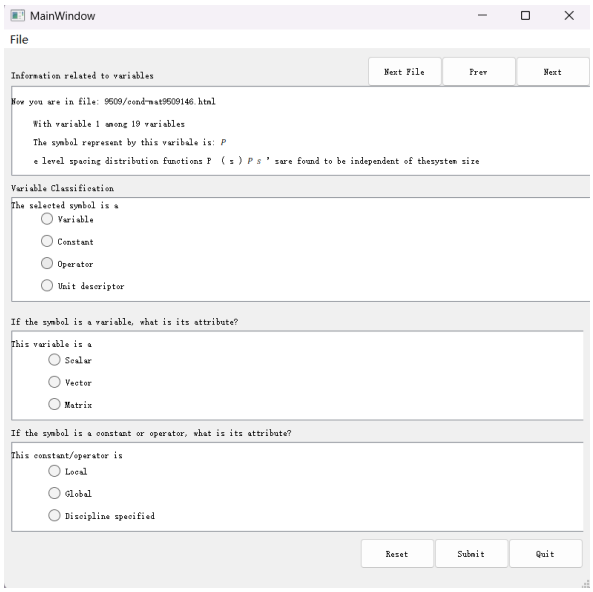

Figure 4: The UI Design of STEM-PoM Labeler

B Additional Experiment Setups

Training Details. In our experiments, we train an LSTM with varying numbers of layers for the mathematical symbol classification tasks. For LLMs fine-tuning, we apply the common parameter-efficient fine-tuning (PEFT) method, LoRA (Hu et al., 2021), to evaluate the model precision performance. Specifically, we set the LoRA rank to 32, batch size to 32, weight decay to 0.01, dropout to 0.1, and learning rate to $1e^{-4}$.

Dataset Split. We divide STEM-PoM into 80%/10%/10% for training/validation/testing sets.

Model Details. For the LSTM model, we use different layer sizes from {128, 256, 512, 1024}. The hidden state size is set to 256, the learning rate is set from {0.1, 0.01, 0.001}, the training epoch is 5, and the batch size is 16. We utilize the Adam optimizer (Kingma, 2014). For LLMs generation, we set the temperature for all models to 0 (Greedy Decoding), top_p to 1.0, frequency penalty to 0, and presence penalty to 0.

C Downstream Math Reasoning

Evaluation Details. We use a unified 3-shot CoT prompt template for all models, provided in Figure 7. Decoding is performed with a temperature of 0 (Greedy Decoding), and we report pass@1 accuracy. Dataset details are outlined below.

- **GSM8K** (Cobbe et al., 2021) is a mathematical dataset comprising 8.5K high-quality, linguistically diverse grade-school math word problems designed for multi-step reasoning. Solutions involve elementary arithmetic operations and require no concepts beyond early algebra. The test set consists of 1,319 unique problems.

- **MATH** (Greiner-Petter et al., 2022) is a dataset of 12,500 challenging competition-level mathematics problems sourced from contests such as AMC 10, AMC 12, and AIME. Each problem is accompanied by a step-by-step solution, enabling models to learn answer derivations and explanations. The test set comprises 5,000 unique problems.

- **OlympiadBench** (He et al., 2024a) is a bilingual, multimodal scientific benchmark comprising 8,476 Olympiad-level math and physics problems, including those from the Chinese college entrance exam. For our evaluation, we use the open-ended, text-only math competition subset, which consists of 674 problems.

Implementation Details. For LLMs generation, we set the temperature for all models to 0 (Greedy Decoding), top_p to 1.0, frequency penalty to 0, and presence penalty to 0.

D Full Analysis on STEM-PoM

Context Length	Variable	Constant	Operator	Unit Descriptor
One Sentence	24.5%	13.2%	10.3%	27.1%
Five Sentence	26.3%	15.6%	14.1%	29.2%
Ten Sentence	28.1%	16.8%	15.5%	30.2%

Table 8: LSTM first-level classification accuracy based on different input context lengths.

Model Performance vs Data Input Lengths. Table 8 displays the classification accuracy of an LSTM model across varying input context lengths across four categories. A trend of increasing accuracy can be observed as the input length increases. For instance, in the Variable category, the accuracy increases from 24.5% for one sentence to 28.1% for ten sentences. Similarly, for the Constant category, accuracy rises from 13.2% for one sentence to 16.8% for ten sentences. The Operator category shows a modest increase from 10.3% to 15.5% as the input length expands. Finally, for the Unit Descriptor category, accuracy grows from 27.1% to

30.2%. These results suggest that longer input data contributes to improved classification accuracy.

Context Length	Overall	Variable	Constant	Operator	Unit Descriptor
			Vanilla Inference		
One Sentence	56.8%	51.5%	53.5%	59.4%	62.4%
Ten Sentences	58.7%	54.5%	53.6%	61.3%	65.1%
Full Manuscript	60.6%	57.2%	56.6%	63.2%	65.2%
			LoRA Fine-tuned		
One Sentence	67.4%	64.8%	67.5%	71.4%	66.1%
Ten Sentences	66.9%	65.4%	66.6%	71.3%	64.5%
Full Manuscript	62.2%	58.4%	62.2%	65.1%	63.2%

Table 9: First-level classification with various context lengths on GPT-3.5 and fine-tuned GPT-3.5.

Fine-tuning vs In-context learning. Table 9 shows the comparison result on main attributes between fine-tuned and directly vanilla-referenced GPT3.5. Notably, the fine-tuned GPT-3.5 model achieves an accuracy of 67.4% in the one-sentence context. However, its performance diminishes as the context length increases, with a noticeable drop to 66.9% for ten sentences and further down to 62.2% for full manuscript-length context. Such diminishing return for fine-tuned models with longer contexts indicates that fine-tuning amplifies sensitivity to the introduction of noisy or less relevant information when longer contexts are involved. The observation also could point to challenges in the fine-tuning process for long-context LLMs, which require more refined techniques to handle context length effectively.

E Case Study on STEM-POM

To further demonstrate the utility and effectiveness of the STEM-POM dataset, we conducted a case study focusing on error-prone classification scenarios involving mathematical symbols with ambiguous or context-dependent attributes. Our main goal for this case study is to evaluate where and why current state-of-the-art language models fail in complex mathematical reasoning tasks.

E.1 Setup

Sub-dataset Selection. Building on the main experimental results, we select a subset of mathematical symbols from STEM-POM dataset—particularly those frequently misclassified—to analyze the model's failure modes.

Evaluation Metrics. We focused on precision accuracy and the frequency of specific errors across the selected symbols. Additionally, we conducted qualitative error analysis to identify consistent failure cases.

Experimental Contexts. Two context lengths were considered for testing:

- **Ten Sentences**: Top Ten sentences selected by human experts surrounding the symbol.
- **Full Manuscript**: Entire input manuscript.

E.2 Results and Analysis

Error Frequency and Common Failure Cases. Table 10 highlights the top ten symbols that LLMs frequently misclassified. The error rates are calculated as the percentage of incorrect predictions across all contexts. Notably, even the best-performing model, GPT-4o, exhibited a significant error rate for symbols with overlapping roles, such as "α" and "Σ".

Symbol	GPT-4o	Claude 3.5	Llama3.1-70B	Mistral-8x7B
α	25%	32%	45%	58%
β	22%	29%	38%	50%
Δ	30%	35%	48%	60%
ψ	28%	33%	44%	57%
θ	26%	31%	42%	55%
Σ	34%	38%	50%	63%
λ	20%	27%	36%	48%
π	18%	25%	34%	46%
μ	24%	30%	41%	54%
ξ	27%	34%	46%	59%

Table 10: Top 10 error-prone symbols and their misclassification rates.

Error Analysis by Symbol Type. The analysis revealed recurring error patterns: **(1) Ambiguity in Context**: Symbols like "α" and "β" were frequently misclassified due to their diverse meanings in different disciplines. **(2) Overlapping Attributes**: "Σ" and "Δ" were often confused between operators and constants depending on context. **(3) Domain-Specific Knowledge Gaps**: Models struggled with symbols like "ψ" and "θ" that require understanding of physics or quantum mechanics.

Performance Comparison. Figure 5 illustrates the misclassification rates for the top ten error-prone symbols across models. GPT-4o consistently outperformed smaller models, but the error rates remained substantial for challenging symbols.

Qualitative Examples. Table 11 provides examples of contextual sentences and the corresponding model predictions. These examples demonstrate

Symbol	Contextual Sentence	Incorrect Prediction (Correct)
α	"The coefficient α controls the rate of decay in the model."	Constant (Correct: Variable)
ψ	"The wave function ψ represents the quantum state of the system."	Variable (Correct: Operator)
Σ	"The summation Σ is over all possible states."	Operator (Correct: Constant)

Table 11: Qualitative analysis of errors for selected symbols.

how insufficient understanding of context leads to systematic errors. Symbols like "α" and "δ" are often misclassified due to ambiguous contextual cues. For example, "α" is misclassified as a constant in differential equations, where it actually serves as a prefix operator given the context.

Impact of Context Length. While longer contexts improve classification accuracy overall, they also introduce noise. For example, in the full manuscript setting, models sometimes overfit on irrelevant context, leading to errors in identifying local operators as global constants.

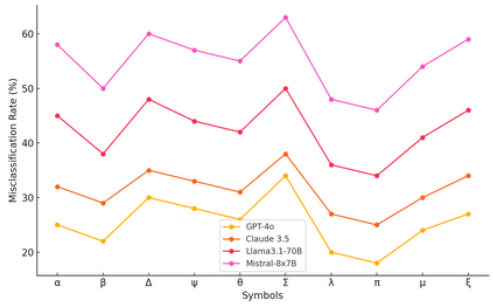

Figure 5: Misclassification rates for the top 10 error-prone symbols across different LLMs.

Through the case study, we conclude that despite the contextual advantages provided by STEM-POM, symbols with ambiguous or multi-faceted roles remain challenging. Larger models like GPT-4o mitigate some of these issues but still fail to handle nuanced sub-attributes effectively.

F Discussions

Impact and Applicability of STEM-PoM on Math Reasoning. In this section, we discuss why classifying mathematical tokens is highly relevant to advancing mathematical reasoning on LLMs.

On the surface level, summarization is a quintessential AI task, and being able to summarize mathematical tokens is a crucial downstream task that has already been held back in several recent research. One notable example is ScholarPhi (Head et al., 2021), where researchers endeavor to make STEM manuscripts more accessible by enhancing a PDF viewer with an annotation tool that summarizes mathematical tokens and equations whenever users click on them. The authors observed that high-quality annotations significantly enhanced users' comprehension of the manuscripts. However, they manually annotated their test sets, as they were uncertain about how annotation accuracy would influence comprehension.

Furthermore, researchers have explored the task of converting mathematical content in manuscripts into computable functions (Greiner-Petter et al., 2022; Greiner-Petter, 2023). In such tasks, where mathematical expressions must be transformed into alternative formats, accurately classifying these tokens is crucial.

Therefore, the classification of mathematical tokens serves as a foundational component for numerous downstream tasks. We developed this dataset to provide researchers with a baseline for evaluating and advancing this critical functionality.

G Model Prompts

For detailed prompts of STEM-PoM and downstream math reasoning tasks, please refer to Figure 6 and 7.

Prompt Template 1: STEM-PoM

You are an advanced AI system specializing in mathematical language processing. Your task is to analyze a given mathematical symbol extracted from a STEM article and its associated contextual content and accurately classify the symbol into its correct primary attribute category based on the provided definitions.

First/Second-level Attribution Categories and Definitions:

1. **Constant**: [Precise definition and examples tailored to the task]
2. **Operator**: [Precise definition and examples tailored to the task]
3. [Additional attribute categories with definitions]

...

Demonstrations for Reference:

Below are three high-quality examples to illustrate the classification process:

<Few-shot example 1>
<Few-shot example 2>
<Few-shot example 3>

Input:

- **Math Symbol**: `\mathcal{X}`
- **Related Content**: Contextual sentences extracted from the original article: `<surrounding sentences from the original article>`

Instructions:

1. Review the provided mathematical symbol and its contextual content.
2. Refer to the main attribute definitions and demonstrations above.
3. Classify the math symbol into its most appropriate primary attribute category.

Output Format:

Return classified attributes only with no additional explanation or formatting.

Figure 6: The prompt design for STEM-PoM.

Prompt Template 2: Downstream Arithmetic Reasoning

You are an advanced AI system specializing in mathematical language processing. Your task is to solve the following math problem efficiently and clearly. Please reason step by step using chain-of-thought to solve the following math problem efficiently and clearly. Please reason step by step and return your final answers.

Problem
<Problem Description>

3-shot Demonstration
Here are three examples similar to your problem, please follow the example to reason step by step and output the answers following the same format.

<Example 1>
<Example 2>
<Example 3>

Your Response:

Figure 7: The prompt design for downstream math reasoning tasks including GSM8K, MATH, and OlypiadBench.