The Role of Math Notation Complexity in ML Conference Publication Success

Linus A. Schneider Matr.-Nr. 6989196 linus.schneider*
Jaisidh Singh Matr.-Nr. 6960379 jaisidh.singh*
Leon Lemke Matr.-Nr. 7077885 leon.lemke*
Anupam Sourav Patra Matr.-Nr. 7003493 anupam.patra*

*@student.uni-tuebingen.de

Abstract

Conference publications can have a significant impact on a machine learning (ML) scientist's career, prompting extensive discussion about the factors that contribute to a paper's acceptance[1]. Although prior research has explored the impact of various attributes of papers [1] on their acceptance, the role of math notation complexity in publication success remains unexplored. To this end, we conduct a data-driven investigation of the relationship between a paper's math notation complexity and conference publication success. We find that math notation complexity plays a subtle but statistically significant role in the acceptance of papers at ICLR. We hope to pave the way for more rigorous investigation of math notational complexity as a part of publication success.

1 Introduction

The relationship between mathematical notation complexity and the success of ML conference publications remains largely unexplored. While complexity may signal novelty, excessive complexity could also hinder readability. Exploring its role may provide insight into how to get accepted into prestigious conferences, which is why we investigate the hypothesis that *more complex math notation in ML papers lead to better chances of acceptance at conferences*.

We first collect relevant real-world data from the ICLR [4] 2023 and NeurIPS [6] 2023 conferences. The mathematical expressions are extracted via LateXML [3] and Pandoc [2], after which we derive a set of features to capture the complexity of mathematical notation. Next, we conduct various statistical analyses which include analysis of correlation of features, separability of accepted and rejected feature populations.

2 Methods

2.1 Dataset Curation

First, we obtain scraped paper acceptance and rejection metadata of the ICLR 2023 and NeurIPS 2023 conferences from OpenReview[7] paired with the papers' titles through the publicly accessible Paperlists[8] repository. Using the arXiv API, we search the title of each paper and, if it's available, download its TEXsource files.

To extract the math expressions, we use LaTeXML [3] and Pandoc [2] to compile the TEXsource to an XML or JSON file respectively. This allows us to parse all kinds of math environments without having to take into account the packages used or custom commands specified.

Due to the TEX-sources often consisting of multiple files, we assume that if only for a single `.tex` file there exists a matching `.bbl` file of the same name, that `.tex` file is to be the main source file to compile from. If multiple matching `.tex` and `.bbl` files exist we skip the paper, as we can't know

[1] https://tinyurl.com/2vdv4xad

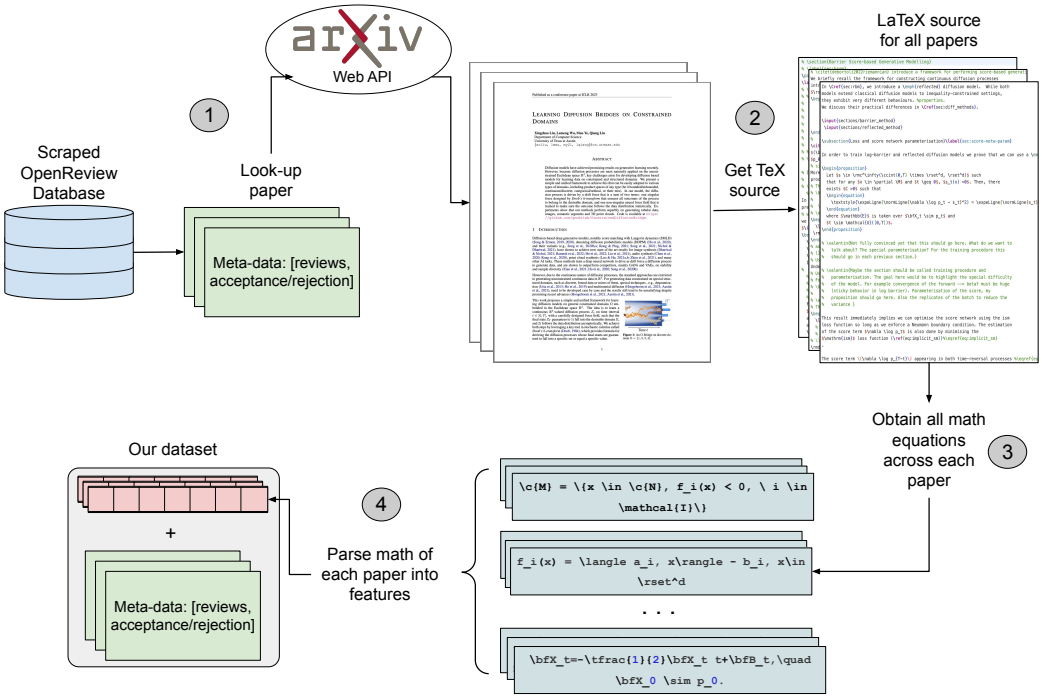

Figure 1: A visual overview of our dataset creation process is shown above.

what to compile. To compile the source, we first attempt to do so using Pandoc, and reattempt using LateXML if that fails.

Following compilation, we extract all math expressions using either the `Math` JSON key for Pandoc output and the `<Math>` XML tag for LateXML output and store them in a `parsed.math` file in a directory named after their paper's OpenReview ID, so that they can be associated with the acceptance data during feature extraction.

We engineer 6 features $(x_1, x_2, \ldots, x_6)$, as described in Table 1 below to represent different aspects of mathematical notation complexity using regular expression matching on the math expressions.

Feature	**Description**
x_1	No. of mathematical expressions as defined by latex (i.e. "$", "`\begin{equation}`")
x_2	Mean no. of new symbols introduced per expression
x_3	No. of distinct mathematical symbols in the paper
x_4	Mean no. of distinct mathematical symbols in an expression
x_5	Standard deviation of the no. of distinct mathematical symbols in an expression
x_6	The maximum no. of unique symbols used in an expression across a paper

Table 1: Notation complexity features used for analysis.

Finally, we create a `.csv` file for each conference, containing columns for each papers OpenReview ID, math complexity features, and conference status. For ICLR, we drop papers with the status *Withdraw*, since they were neither rejected nor accepted and *Desk Reject* papers, since they were not reviewed. The status *Top-25%*, *Top-5%* and *Poster* are considered *Accepted*. For NeurIPS, *Poster*, *Spotlight* and *Oral* are considered *Accepted*. This gives us the following class sizes for ICLR 2023: *Reject* **(435 papers)**, *Accept* **(522) papers**. For NeurIPS, the sizes are *Reject* **(149 papers)**, *Accept*

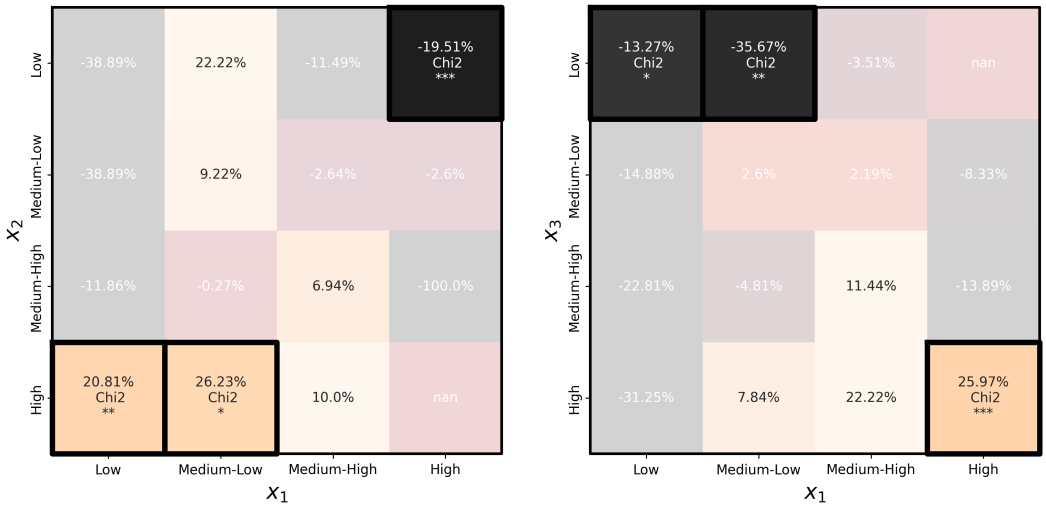

Figure 2: Interactions of quartiles of different features and their influence on acceptance, along with the test that was done as well as if it was statistical significant ($p \leq 0.001$: ***, 0.01 : **, 0.05 : *). `NaN` means that there were no papers with that combination in the dataset.

(2771 papers). Due to the large class imbalance in our NeurIPS dataset, we only consider the ICLR 2023 dataset during analysis, for which the accepted/rejected classes are better balanced.

2.2 Analysis

We initially apply several hypothesis tests to evaluate if more complex math notation leads to higher acceptance rates. The relationship of each feature with paper acceptance is examined using three different approaches: (i) the Student's T-test, (ii) Welch's T-test, and (iii) Mann-Whitney U-test. We also calculate Cohen's D-effect size with bootstrap confidence intervals, that quantifies the magnitude and stability of the differences between features of accepted and rejected papers.

Next, we analyze how features work together to impact acceptance via a two-stage approach. First, we identify potentially important feature combinations by the relation of their joint variation with acceptance. Second, we divide each feature into 4 equally sized quartiles based on its value (Low, Medium-Low, Medium-High, High). For each promising combination of features from stage 1, we divide all papers into 16 groups based on all quartile combinations, and calculate the relative change in acceptance rate compared to the overall mean for each combination. Then, we test whether each combination's acceptance pattern significantly differs from random chance via Fisher's exact test for sample sizes <30 and chi-square (χ^2) tests for larger sample sizes.

3 Results

The hypothesis tests and Cohen's D for measuring effect size show that all features, with exception of x_6 (maximal representational complexity), are statistically significant with regards to acceptance, however, effect sizes are small to negligible. This is shown in Table 2. Furthermore we can see that all features, except for x_2, have a positive influence on acceptance.

Through the analysis of the joint variation of feature combinations with acceptance, we identify multiple candidates for interaction analysis. Among them, the interaction between x_1 and x_3 shows a strong significance with an F-statistic of 16.487 (p = 5.301e-05), followed closely by the interaction between x_1 and x_2 with an F-statistic of 16.026 (p = 6.731e-05), indicating highly significant relationships in how these feature pairs jointly influence acceptance rates. A more detailed analysis of their interaction, as shown in Figure 2, gives more detailed insight into their relationship. Papers with a large number of expressions (x_1) and a high number of distinct mathematical symbols (x_3) show a 25.97% higher acceptance rate ($p \leq 0.001$), while those using a medium-low ($p \leq 0.01$) to low ($p \leq 0.05$) number of distinct symbols (x_3) and a low number of math expressions (x_1), show

a 35.67% to 13.27% lower acceptance rate. In an inverse fashion, papers with a low ($p \leq 0.01$) to medium low ($p < 0.05$) number of expressions (x_1) and a high number of new symbols introduced per expression (x_2), show a 20.81% to 26.23% higher acceptance rate and papers with a large number of expressions (x_1) but a low number of new symbols introduced per expression (x_2), show a 19.51% reduction in acceptance ($p \leq 0.001$).

Feature	Means (A/R)	Cohen's D [95% CI]	Effect	t-test	Welch	M-W
x_1	547.55/424.94	0.271 [0.147, 0.387]	small†	0.000*	0.000*	0.000*
x_2	0.20/0.23	-0.253 [-0.386, -0.125]	small†	0.000*	0.000*	0.000*
x_3	72.24/64.34	0.260 [0.142, 0.381]	small†	0.000*	0.000*	0.000*
x_4	3.29/3.00	0.205 [0.083, 0.329]	small†	0.002*	0.002*	0.002*
x_5	3.99/3.76	0.174 [0.041, 0.295]	negligible†	0.008*	0.008*	0.004*
x_6	22.95/22.20	0.111 [-0.016, 0.234]	negligible	0.088	0.092	0.127

Table 2: Hypothesis tests comparing groups A (522 samples) and R (435) across multiple features. †: Effect direction is stable across bootstrap samples, *: $p < 0.05$.

4 Discussion

This study investigates math notation complexity's impact on paper acceptance at a machine learning conference. Our main contributions lie in assembling the novel dataset, engineering features to represent math notation complexity, and using statistical analysis to provide initial empirical evidence in this previously unexplored domain.

We recognize that our dataset might contain bias due to availability of papers from ICLR 2023 on arXiv. Out of 4955 papers submitted only 1537 were available on arXiv as TEXsource-files. A remedy to this can be to reverse engineer the source through TEXOCR tools like Mathpix [5] directly. We can also enhance the math extraction math and feature engineering, by using different parsing tools, regular expression matching, and implementing a recursive approach for properly handling subscript symbols during feature extraction. While our custom features may capture certain aspects of math notation complexity, it would be useful to validate them via a survey of reviewers.

Our study shows that although there is a statistically significant effect of math notation complexity on conference acceptance, it is subtle. There does, however, emerge preliminary evidence towards best practices for math notation in writing scientific papers. Namely, more expressions built with larger symbol vocabularies (high x_1 with high x_3) are statistically better off in terms of acceptance. This is also true for fewer expressions packing a high bit-rate of math (low x_1 with high x_2).

5 Statement of Contributions

The contributions per team member are given as follows. LS managed organizational tasks like meetings and todos, dataset metadata, and `tex` source extraction. LL focused on math extraction and dataset creation. JS and ASP collaborated on feature extraction, while JS handled correlation analysis. JS also worked with LS on hypothesis testing. LS conducted interaction tests and created plots with LL. Everyone participated in dataset cleaning and report writing. Additional analyses of using Machine Learning for acceptance prediction (LL), k-means clustering (ASP), and box plots (ASP), were ultimately not used in the final project.

References

[1] Mikhail Skorikov and Sifat Momen. "Machine learning approach to predicting the acceptance of academic papers". In: *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. 2020, pp. 113–117. DOI: 10.1109/IAICT50021.2020.9172011.

[2] John MacFarlane. *Pandoc*. https://github.com/jgm/pandoc/releases/tag/3.1.3. Version 3.1.3. 2023.

[3] Burce R. Miller and Deyan Ginev. *LaTeXML*. https://github.com/brucemiller/LaTeXML/releases/tag/v0.8.8. Version 0.8.8. 2024.

[4] *International Conference on Learning Representations*. Accessed: 10-02-2025. URL: https://iclr.cc.

[5] *Mathpix*. Accessed: 10-02-2025. URL: https://mathpix.com/.

[6] *Neural Information Processing Systems*. Accessed: 10-02-2025. URL: https://neurips.cc.

[7] *OpenReview*. Accessed: 10-02-2025. URL: https://openreview.net.

[8] *OpenReview PaperLists*. Accessed: 10-02-2025. URL: https://github.com/papercopilot/paperlists.