



MathemaTikZ: A Benchmark for K-12 Mathematical Diagram Generation

Rizwaan Malik
rizmalik@stanford.edu
Stanford University
Graduate School of Education
Stanford, California, United States

Ritika Kacholia
ritikak@stanford.edu
Stanford University
Graduate School of Education
Stanford, California, United States

Rebecca Li Hao
rhao@stanford.edu
Stanford University
Graduate School of Education
Stanford, California, United States

Dorottya Demszky
ddemszky@stanford.edu
Stanford University
Graduate School of Education
Stanford, California, United States

Abstract

Diagrams play a fundamental role in mathematics education, serving both as essential components of mathematical problems and as powerful scaffolding tools to support student comprehension. While AI tools have shown promise in supporting teachers with lesson preparation, especially with text-based mathematical content, they still struggle with reliably generating visual diagrams. Our work makes two main contributions: (1) We introduce MathemaTikZ, a dataset derived from the Illustrative Mathematics curriculum, comprising 3,793 mathematical diagrams paired with their natural language descriptions, problem contexts, and TikZ implementations. These span the full range of diagrams utilized in the K12 math curriculum. (2) We conduct comprehensive baseline evaluations using state-of-the-art language models (GPT-4o, Claude 3.5 Sonnet, and Gemini 2.0 Flash) to assess current capabilities in mathematical diagram generation. Our findings reveal that even the best-performing models achieve a 73.9% success rate in accurately generating mathematical diagrams, with performance varying significantly across different types of visualizations. Through detailed error analysis, we identify four key challenge areas that future work should address: spatial reasoning and element placement, adherence to geometric constraints, pedagogical knowledge of mathematical diagrams, and preservation of mathematical relationships. Our results establish baselines for mathematical diagram generation and highlight critical areas for improvement in making AI tools more effective for mathematics education.

CCS Concepts

• **Social and professional topics** → **K-12 education**; • **Human-centered computing** → **Visualization systems and tools**.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.


L@S '25, Palermo, Italy

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0633-2/24/07
<https://doi.org/10.1145/3657604.3664698>

Keywords

Mathematics Education, Large Language Models, Visual Generation, Curriculum Development

ACM Reference Format:

Rizwaan Malik, Rebecca Li Hao, Ritika Kacholia, and Dorottya Demszky. 2025. MathemaTikZ: A Benchmark for K-12 Mathematical Diagram Generation . In *Proceedings of the Twelfth ACM Conference on Learning @ Scale (L@S '25)*, July 22–23, 2025, Palermo, Italy. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3657604.3664698>

1 Introduction

The challenge of automated diagram generation is particularly acute in the context of curriculum adaptation and customization. While Large Language Models (LLMs) have shown promise in generating text-based mathematical content and practice problems [17], their capability to generate accurate and pedagogically sound visual aids remains relatively unexplored. This gap is particularly significant for the customization of high-quality curriculum materials like Illustrative Mathematics, where visual representations are integral to problems, explanations, and student scaffolding.

Recent work has explored various approaches to mathematical diagram generation, from using domain-specific languages [25] to multi-agent architectures [16]. However, there has been no systematic evaluation of how different factors—such as description quality, syntax guidance, and model architecture—affect diagram generation accuracy. Understanding these relationships is crucial for developing effective educational tools and identifying key areas for improvement.

Our work makes two main contributions. First, we introduce **MathemaTikZ**, a dataset of 3,793 mathematical diagrams derived from the Illustrative Mathematics (IM) curriculum, paired with their natural language descriptions, problem contexts, and TikZ implementations. This dataset spans the full range of diagrams utilized in K-12 mathematics education, providing a comprehensive resource for developing and evaluating mathematical visualization systems. The dataset is available to researchers on request by filling in the form at bit.ly/mathematikz.

Second, we present a **systematic evaluation of mathematical diagram generation** across six experimental conditions, testing

three state-of-the-art language models (GPT-4o, Claude 3.5 Sonnet, and Gemini 2.0 Flash). We examine how performance varies with different quality levels of diagram descriptions (from basic alt-text to human enhanced descriptions) and with the addition of TikZ syntax guidance. Our findings reveal that even in optimal conditions—using human enhanced descriptions and syntax primers—the best-performing models achieve a 73.9% success rate in generating mathematically accurate diagrams. Through detailed error analysis, we identify four key challenges: spatial reasoning and element placement, adherence to geometric constraints, pedagogical knowledge of mathematical diagrams, and preservation of mathematical relationships.

2 Related Work

2.1 AI for Curriculum Customization and Scaffolding

High-Quality Instructional Materials (HQIM) have demonstrated the potential to yield learning gains in mathematics when implemented effectively [1, 12]. However, some studies show that actual improvements from HQIM adoption at the district-level are often modest [4, 5]. One explanation for this discrepancy is that teachers frequently do not implement the materials with fidelity, for reasons including a lack of alignment to local contexts and the presence of substantial numbers of students performing below grade level [4]. In these situations, teachers may need to adapt or supplement high-quality curricula to meet diverse student needs while still maintaining curricular coherence and integrity [17].

LLMs present an opportunity to address these adaptation challenges. Rather than discarding the structure or rigor of HQIM, LLM-based systems can preserve core pedagogical design while creating additional scaffolds that target specific learner gaps. For instance, Malik et al. [17] investigated whether LLMs could generate “warm-up” exercises to bridge students’ prior knowledge deficits and support grade-level content. Their framework decomposed the teacher’s scaffolding role into three stages—observing students’ needs, formulating a strategy, and implementing that strategy with concrete tasks. With access to curriculum materials and an “expert prompt,” the model produced warm-ups rated highly by educators for alignment with instructional objectives and for accessibility for students working below grade level.

Student-facing applications are also emerging. Sun et al. [23] describe ScaffoldiaMyMaths, which integrates AI-driven scaffolding into an existing elementary math platform. The system provides real-time, context-specific prompts (e.g., fraction strips, step-by-step hints) based on a live analysis of students’ interactions with problems. Preliminary evidence suggests this personalized support can boost both engagement and comprehension, much like a human tutor but at classroom scale.

2.2 Diagrams in Math Education

A key gap in the effectiveness of AI-driven curriculum-related applications has been the inability to generate math diagrams [21]. Diagrams play a critical role in mathematics education, both as essential components of mathematical problems and as powerful scaffolding tools to support student comprehension [6, 22]. They are used as conceptual frameworks to support student thinking, such

as number lines to support arithmetic [10] and fractions [2], tape diagrams for proportional reasoning [19], and balance diagrams to understand equivalence [24]. Diagrams have been shown to be effective when they conceptually align with the math problem [11]. The effectiveness of diagrams to support student math achievement varies for students of different attitudes and achievement [8], underscoring the importance of adapting mathematics instruction and diagram use to individual student needs.

2.3 Automated Diagram Generation

Several approaches have been used to generate accurate mathematical diagrams. Early text-to-image models made rapid progress in producing photorealistic images from prompts [20], yet struggled with the structured layouts and exact labeling required for mathematical and scientific diagrams [27]. For instance, conventional diffusion-based models often fail to maintain specific geometric constraints, annotate complex relationships, or reproduce text accurately, a critical shortfall in education contexts where factual correctness supersedes aesthetics (e.g., geometry proofs, physics flowcharts) [7, 28].

Pre-LLM approaches explored specialized frameworks, including domain-specific languages (DSLs), to create structured environments for minimizing model generation errors. Ye et al. [25] introduced Penrose, which uses constraint-based specification to automatically optimize diagram layouts from high-level mathematical statements, enabling diverse visual representations (e.g., Euclidean vs. hyperbolic) from the same content.

With the rise of LLMs, research has shifted to code generation based on evidence that these models also benefit from clearly defined syntax and structure when generating diagrams. Jain et al. [13] show that LLMs can learn to generate diagrams using DSL based on the Penrose framework when provided with few-shot examples. Likewise, Belouadi et al. [3] introduced DaTikZ and the broader AutomaTikZ framework, which focus on generating TikZ code from textual descriptions. Their dataset of 120k paired examples (code plus captions) enabled fine-tuning LLMs specifically for vector-graphics creation. By leveraging these DSLs, the models achieved significantly higher fidelity than generic text-to-image models like DALL-E or Stable Diffusion, especially for scientific or mathematical content.

Recent multi-stage pipelines further illustrate LLM-based diagramming strategies. SciDoc2Diagrammer-MAF [18] refines Graphviz or TikZ code iteratively based on long scientific texts, reducing hallucinations common in direct text-to-image models. DiagrammerGPT [26] employs an LLM-based “planner-auditor” loop to outline diagram structure, followed by a diffusion renderer. The LLM enforces semantic correctness (e.g., arrow connections, proper labeling), resulting in diagrams that are more accurate and visually coherent than naive diffusion models can achieve. Lee et al. [15] use LLMs for SVG generation and evaluation for mathematical diagrams and hints, such as those used by applications like Khan Academy and IXL learning.

An important open question in LLM mathematical diagram generation and evaluation is whether LLMs are able to understand,

evaluate, and properly generate spatial and mathematical information. Currently, evidence suggests that language models have trouble understanding mathematical diagrams and geometric reasoning [14, 27], motivating the need to better understand and address these challenges.

Across these efforts, vector-based approaches remain the most effective at controlling geometry and labels precisely. This is central to educational scenarios where a slight positional or labeling error can yield incorrect meaning for students. Open challenges include ensuring total fidelity in multi-step or highly specialized diagrams (e.g., circuit schematics, intricate 3D geometry), enabling real-time interactive generation, and developing standardized benchmarks for diverse visualization tasks. Despite these methodological advances in diagram generation, work is still limited on generating pedagogically-sound diagrams for use in K-12 mathematics education, further motivating our work.

3 MathemaTikZ Dataset

3.1 Data Source

Our dataset, MathemaTikZ, is derived from Illustrative Mathematics (IM), a comprehensive problem-based core curriculum that serves over four million K-12 students across the US. The IM curriculum is open-source and available online. The curriculum is particularly known for its emphasis on visual representations, incorporating diverse types of mathematical diagrams that support student learning through carefully designed sequences of activities. The curriculum received the highest possible ratings from EdReports [9] across all evaluation dimensions, including focus, coherence, rigor, and usability, which makes it an ideal source for high-quality mathematical visualizations.

Each lesson in the IM curriculum includes various diagrams that serve multiple pedagogical purposes: supporting concept introduction, demonstrating problem-solving approaches, and providing opportunities for student practice. These visuals are particularly crucial in supporting diverse learners, including multilingual students and students with disabilities, through careful attention to accessibility and multiple representations of mathematical concepts.

3.2 Data Processing

IM shared an original dataset of 32,007 rows containing diagrams used across the curriculum. We filtered this dataset to include only images that are associated with valid images in the current student materials of the curriculum. This filtering approach ensures all images refer to complete, final images rather than drafts, and preserves the important contextual information needed for understanding each diagram. After this initial filtering, our dataset was reduced to 3,793 unique mathematical diagrams.

Each entry in MathemaTikZ includes:

- Unique identifiers (lesson-id, task-id)
- Contextual information (context) containing the original problem statement that the image appears in
- Image descriptions (original-description, revised-description)
- Technical implementation (tikz-code)
- Visual content (image-url)

One challenge in preparing the dataset was the insufficient detail in many of the original image descriptions. To address this issue, we generated enhanced descriptions by providing a language model with both the original description and the TikZ code, instructing it to create more comprehensive and mathematically precise descriptions. Henceforth, we often refer to the original description as IM alt-text and the revised description as revised alt-text for brevity.

3.3 Dataset Characteristics

Each diagram in MathemaTikZ is represented by both its visual form and its TikZ code implementation, allowing for programmatic analysis and modification. Analysis of component lengths revealed significant variation across the dataset. TikZ code segments averaged 778.20 characters (Median: 628.00, SD: 685.95), ranging from 119 to 11,028 characters. Context information was substantially longer at 2,480.40 characters on average (Median: 2,010.00). IM alt-texts were notably brief, averaging just 73.21 characters (Median: 58.00), while our revised alt-texts were significantly more detailed at 461.08 characters on average (Median: 447.00).

An interesting aspect of our dataset is the presence of **custom TikZ functions** developed specifically for the IM curriculum. These custom functions include code to create number lines, hanger diagrams, division representations, and various geometric shapes and manipulatives. We estimate that approximately 9% of the diagrams in the dataset utilize these specialized commands.

The diagrams span a wide range of mathematical concepts and visualization types. To systematically categorize these diagrams, we first generated embeddings of the diagram descriptions using TF-IDF vectorization and applied hierarchical clustering to identify natural groupings. Manual inspection of these clusters revealed recurring patterns of mathematical visualization types, which we used to develop a comprehensive categorization schema. We then used regular expressions and pattern matching to estimate the distribution of diagrams across these categories, as shown in Table 1.

Figures 1, 2 and 3 illustrate a range of the provided diagrams with their associated alt-text.

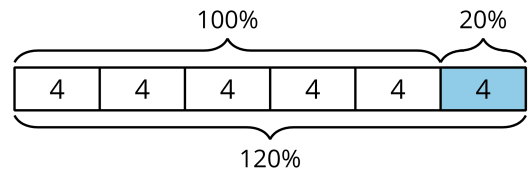


Figure 1: IM alt-text: A tape diagram. Four white sections, labeled 100 percent. With a smaller blue section labeled 25 percent that extends past the 100 percent.

4 Modeling and Evaluation

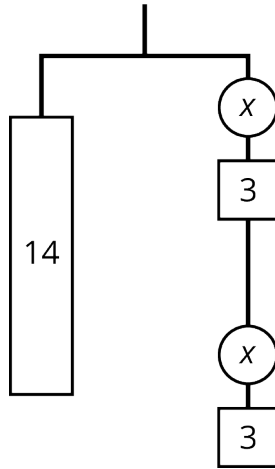
4.1 Experimental Conditions

Initial exploration with diagram generation revealed two primary challenges. First, some descriptions from IM lacked sufficient detail for accurate diagram recreation. For instance, an alt-text might state, “A number line” but omit crucial features such as scale markings

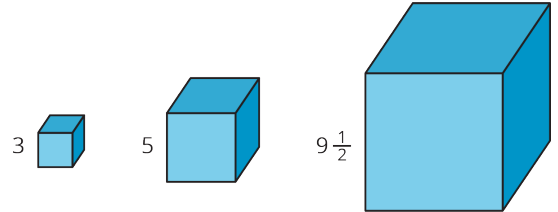
Table 1: Distribution of Mathematical Diagram Types

Category	Diagram Types
Geometric Constructions (60.4%)	Basic shapes (triangles, rectangles, circles), complex shapes (polygons, irregular shapes), properties (congruence, similarity), transformations (reflections, rotations), symmetry
Measurement & Analysis (52.7%)	Angles (acute, obtuse, complementary), lines and segments (parallel, perpendicular, intersecting), line segments and rays, distance and length measurements
Coordinate Systems & Functions (52.2%)	Coordinate planes, grids and graph paper, function graphs (linear, quadratic, exponential), graph transformations
Number Sense & Operations (41.9%)	Number lines, fraction models, place value representations, operation models (multiplication/division, addition/subtraction, integer operations)
Algebra & Expressions (26.3%)	Equations, expressions, variables, inequalities, substitution models, algebraic relationships
Data & Statistical Representations (23.7%)	Distribution plots (histograms, box plots), relationship plots (scatter plots, line graphs), categorical data (bar graphs, pie charts), statistical measures visualizations
Mathematical Models (21.1%)	Measurement tools (rulers, protractors, clocks), visual models (tape diagrams, ten frames, number bonds, area models, arrays), problem-solving diagrams (hanger diagrams, bar models, pattern blocks)
Spatial & 3D Geometry (14.0%)	3D shapes (cubes, cylinders, spheres), cross-sections and projections, nets and unfolded views, volume and surface area models

Note: The distribution of these categories are based on our analysis of 3,793 diagrams. Of these, 89.8% (3,405) were classified into multiple categories, with an average of 3.00 categories per diagram. Only 0.7% (27) diagrams could not be categorized. The percentages sum to more than 100% as diagrams could be assigned to multiple categories.

**Figure 2: IM alt-text: Balanced hanger diagram, left side, rectangle 14, right side, circle x, square 3, circle x, square 3.**

or directional arrows. Second, even when models received detailed descriptions, they often produced non-compiling TikZ code due to syntax errors (e.g., missing packages, incorrect environment

**Figure 3: IM alt-text: Three cubes of different sizes: first cube has side length 3, second cube side length 5, and third cube has side length 9 and 1/2.**

usage). To systematically address these challenges, we designed six experimental conditions:

- (1) **IM Alt-Text Only:** Using the original IM alt-text without additional guidance.
- (2) **IM Alt-Text + TikZ Primer:** Augmenting the IM alt-text with a TikZ syntax guide (henceforth, TikZ primer) to mitigate syntax errors when compiling. The guide includes examples of (1) necessary LaTeX packages or libraries and (2) drawing basic shapes with labeled edges/vertices/angles.

The code and a visualization of the guide is included in the supplement.¹

- (3) **LLM-Generated Alt-Text Only:** Generating alt-text via an automated LLM-based pipeline, incorporating the problem statement and alt-text provided by IM. The LLM was prompted to describe geometric relationships, measurements, and spatial configurations with mathematical precision. Our prompt can be found in the supplement.²
- (4) **LLM-Generated Alt-Text + TikZ Primer:** Supplementing the LLM-generated alt-text with the TikZ primer.
- (5) **Human-Enhanced Alt-Text Only:** Three authors independently reviewed and refined the LLM-generated alt-text against the original IM image and problem statement through a structured evaluation process. Each evaluator first assessed the alt-text independently using a standardized rubric focusing on three key aspects: (1) mathematical accuracy of relationships and constraints, (2) completeness of geometric descriptions, and (3) correctness of numerical labels and annotations. Discrepancies between evaluators were resolved through consensus meetings, where differences were discussed and reconciled to produce a final, validated version of the LLM-generated alt-text. This human enhancement was only applied to the test set (see below).
- (6) **Human-Enhanced Alt-Text + TikZ Primer:** Combining the consensus-validated alt-text with the TikZ primer.

All six experimental conditions were evaluated across three state-of-the-art LLMs: GPT-4o, Claude 3.5 Sonnet, and Gemini 2.0 Flash, selected to represent different training paradigms in the commercial LLM landscape (i.e., different approaches to model architecture, training data selection, and instruction-tuning strategies that shape how each model processes and generates text).

4.2 Evaluation Process

4.2.1 Test Set Construction. We constructed our test set through a stratified sampling of the Illustrative Mathematics Grade 7 curriculum. We selected grade 7 as our focus due to its comprehensive coverage of units spanning geometry, statistics, and algebra, which necessitates a diverse array of diagram types. We selected six diagrams from each of the nine units in Grade 7, resulting in an initial sample of 54. We then excluded diagrams not originally created with TikZ (e.g., real-world images like maps or photographs), yielding a final test set of 50 programmatically generated diagrams. This approach ensured that we sampled both a variety of mathematical concepts and diverse visualization types, while maintaining the focus on TikZ-based figures.

4.2.2 Scoring Methodology. For each diagram in the test set, we generated TikZ code under all six conditions with each of the three LLMs, yielding 18 total outputs per diagram. The resultant 900 diagrams (50 diagrams \times 18 condition-model combinations) were then shuffled and anonymized — meaning, the evaluators did not know which model or condition produced each output.

Each output was independently scored by three evaluators based on whether it faithfully conveyed the same mathematical information as the original IM diagram. Specifically, each evaluator asked:

“Does this generated diagram preserve the critical mathematical details (e.g., numeric values, geometric constraints, distribution patterns) present in the reference diagram?”

If minor stylistic variations (e.g., label placement, minor color differences) did not impact the mathematical integrity of the diagram, the evaluator still deemed it acceptable, as shown in Figure 4. Conversely, outputs that omitted key elements, misrepresented proportions, or failed to compile into a viewable figure were marked as unsuccessful. The evaluators conducted their assessments independently, then resolved any discrepancies through discussion. Final scores were assigned by consensus.

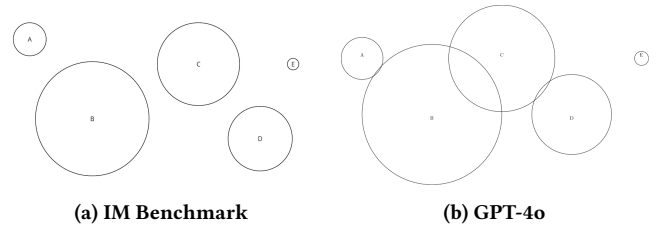


Figure 4: Example of an acceptable stylistic variation. While the generated diagram (b) shows overlapping circles compared to the benchmark (a), the core mathematical properties—the relative sizes of the five circles and their measurability for diameter and circumference comparison—are preserved.

5 Results and Analysis

5.1 Overall Performance

Table 2 presents the comprehensive performance metrics across all experimental conditions. The results demonstrate that both the quality of image descriptions and the inclusion of the TikZ primer impacted the models’ ability to generate mathematically accurate diagrams.

The compilation rates were generally high across all conditions and models, ranging between 80-96%. **The TikZ primer, and enhancements to the image description generally improved compilation rates.** However, compilation is a low bar compared to the success rate—our key metric.

Claude 3.5 Sonnet achieved our highest success rate of 73.9% using only the human-enhanced alt-text, followed by GPT-4o (71.7%) and Gemini (60.5%) with the human-enhanced alt-text and TikZ primer. The quality of the alt-text had a substantial impact on models’ success rates. The success rate increased substantially from approximately 12-15% with original IM alt-text to 24-27% with LLM-generated descriptions, and further improved to 58-74% with human-enhanced descriptions across all models. This finding demonstrates the crucial role of precise and detailed image descriptions, as illustrated in Figure 5. In this example, the original IM description failed to specify that a should be larger than 2, resulting in equal-sized bars. The human-enhanced description explicitly defined this inequality, enabling models to generate a mathematically accurate representation.

¹OSF Supplement: TikZ Primer Code and Visualization

²OSF Supplement: Prompt to Generate AI Image Description Alt-Texts

Table 2: Model Performance Across Experimental Conditions

	GPT-4o		Claude 3.5 Sonnet		Gemini 2.0 Flash	
	Compile (%)	Success (%)	Compile (%)	Success (%)	Compile (%)	Success (%)
IM Original Alt-Text	84.0	11.9	80.0	15.0	80.0	12.5
+ TikZ Primer	86.0	14.0	84.0	21.4	82.0	12.2
LLM-Generated Alt-Text	92.0	26.1	88.0	27.3	82.0	24.4
+ TikZ Primer	96.0	27.1	90.0	26.7	80.0	17.5
Human-Enhanced Alt-Text	92.0	63.0	92.0	73.9	92.0	58.7
+ TikZ Primer	92.0	71.7	86.0	65.1	86.0	60.5

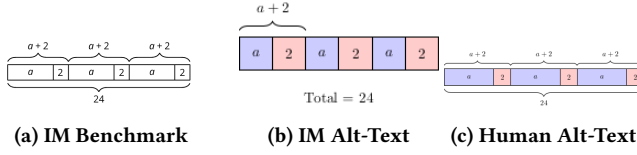


Figure 5: Impact of description quality on mathematical accuracy in a tape diagram (unit 6). The original description did not specify the relationship between a and 2, leading to an incorrect visual implication that $a = 2$. The human-enhanced description explicitly defined this relationship, resulting in correct proportions.

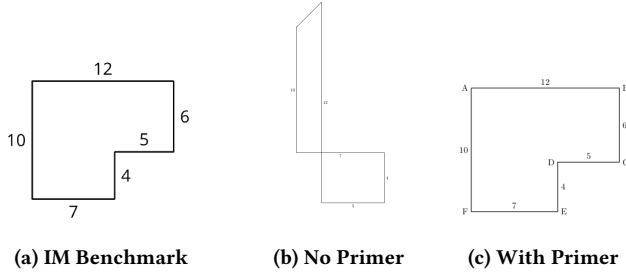


Figure 6: Impact of the TikZ primer on geometric accuracy with the human-enhanced alt-text. The L-shaped hexagon (unit 1), generated by Gemini 2.0, shows marked improvement with the primer. While additional vertex labels were added, they do not compromise the mathematical integrity of the diagram.

With the TikZ primer, 5 out of 9 cases show increased success rates, but apart from those instances with IM alt-texts, these improvements are relatively small. The TikZ primer’s impact on the compile and success rate is particularly evident in complex geometric figures, as shown in Figure 6. When provided with the primer, models demonstrated improved ability to handle intricate shapes while maintaining mathematical accuracy.

GPT-4o, Claude 3.5, and Gemini 2.0 were able to **generate high-quality diagrams that retained mathematical accuracy across all diagram types and units in the IM curriculum**. To further illustrate the models’ capabilities across different mathematical concepts, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12,

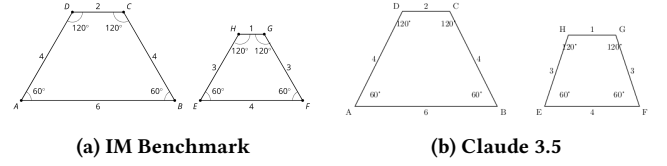


Figure 7: The problem statement (unit 1) asks students to measure angles and work with quadrilaterals. Claude’s diagram maintains mathematical accuracy throughout all the labels, but without the TikZ primer, is unable to draw angle arcs.

Figure 13, Figure 14 present a series of LLM-generated diagrams alongside their IM benchmark counterparts, all produced under the highest-performing experimental conditions (human enhanced alt-texts and TikZ primers for some examples).

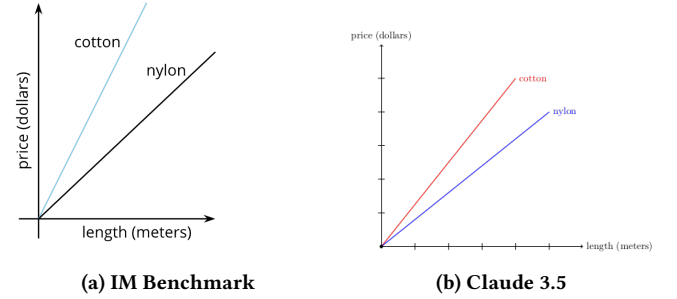
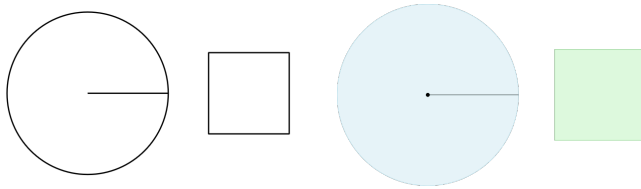


Figure 8: The problem statement (unit 2) compares the slopes of the two lines. Claude’s diagram includes all the correct labels, and intentionally does not include any units on the axes or grid-lines because of the description.

5.2 Performance by Diagram Type

Following the 9 units in IM’s curriculum for Grade 7, Table 3 shows the compilation and success rate for each unit.

Performance varied considerably across different IM units representing different categories of mathematical diagrams, as shown in Table 3. Units 2 and 5 had 100% compile rate, consisting of Cartesian plots, graphs, number lines, and a discrete units diagram for ratios. Units 5 and 4 had the highest success rates, consisting of



(a) IM Benchmark

(b) GPT-4o

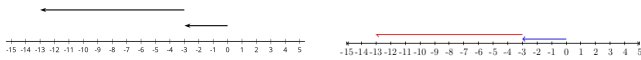
Figure 9: The problem statement (unit 3) compares the area of the circle to square when the radius equals the square’s side length. OpenAI’s visual maintains mathematical accuracy, while also being more visually appealing.



(a) IM Benchmark

(b) GPT-4o

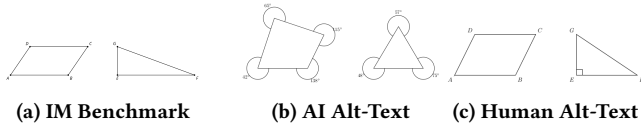
Figure 10: The problem statement (unit 4) compares price to percentages on a double number line. OpenAI’s diagram perfectly matches the benchmark and also maintains consistent significant figures.



(a) IM Benchmark

(b) Gemini 2.0

Figure 11: The problem statement (unit 5) uses a number line to explore the start and end position of a sea animal. Gemini’s diagram perfectly matches the benchmark.



(a) IM Benchmark

(b) AI Alt-Text

(c) Human Alt-Text

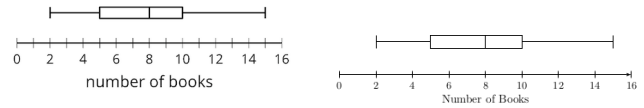
Figure 12: The problem statement (unit 7) asks students to identify complementary and supplementary angles. The AI alt-text hallucinated angle values and the configuration of each shape. (b) and (c) were generated by Claude 3.5.

single/double number lines, tape diagrams, and graphs. Units 7 and 1 had the lowest success rates, consisting of complex 2D and 3D geometric shapes, often including labels and angles.

5.3 Error Analysis

To complement our quantitative findings, we conducted a preliminary qualitative case study of the errors we observed. While not exhaustive, this analysis highlights recurring patterns that could inform future work. Our observations revealed four primary categories of errors that persisted across models and conditions:

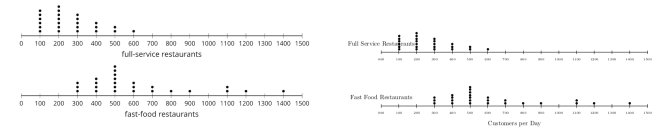
5.3.1 Spatial Reasoning and Element Placement. Models frequently struggled with the precise placement of elements within diagrams. In Figure 15, while the models generated correct sizes and shapes,



(a) IM Benchmark

(b) Claude 3.5

Figure 13: The problem statement (unit 8) explores mean vs median using a box plot. Claude’s diagram is mathematically accurate and incredibly well formatted.



(a) IM Benchmark

(b) Gemini 2.0

Figure 14: The problem statement (unit 9) compares data across two dot plots. Gemini’s diagram maintains two dot plots with accurate data points presented. There is a slight formatting overlap with title of the top dot plot, but that does not detract from the mathematical accuracy.

Table 3: Results Across Units for Human-Enhanced Alt-Text

Illustrative Mathematics G7 Units	Compile (%)	Success (%)
1: Scale Drawings	80.0	50.0
2: Proportional Relationships	100.0	77.8
3: Measuring Circles	94.4	64.7
4: Proportional Relationships and Percentages	86.7	84.6
5: Rational Number Arithmetic	100.0	90.0
6: Expressions, Equations, Inequalities	88.9	56.3
7: Angles, Triangles, Prisms	94.4	38.2
8: Probability and Sampling	86.1	54.8
9: Putting it All Together	76.7	78.3

these shapes were sometimes placed off of the coordinate plane or overlapped with each other. Scale drawings had the third lowest success rate with human-enhanced alt-texts at 50.0% (Table 3).

5.3.2 Adherence to Geometric Constraints. A second major category of errors involved violations of fundamental geometric constraints. Figure 16 demonstrates a case where the angles 35° , h° , and the vertical angle for g° were meant to be supplementary, but failed to maintain the 180-degree sum relationship, and the relationship between g° and its vertical angle, rendering the diagram mathematically incorrect despite its superficial visual similarity to the target. These diagrams were generated with human-enhanced alt-texts and no TikZ primer.

In 3D diagrams, there are many examples of incomplete or not well-formed 3D shapes, such as Figure 17a. Angles, Triangles, and Prisms had the lowest success rate for human-enhanced alt-text at 38.2% (Table 3).

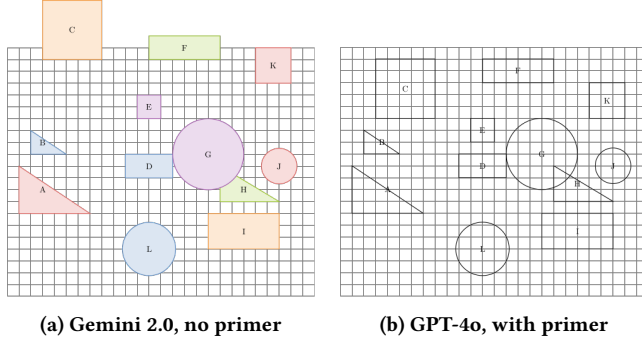


Figure 15: When drawing diagrams, models often struggled to place objects accurately. In both of these examples, shapes were placed off the grid and overlapping with each other.

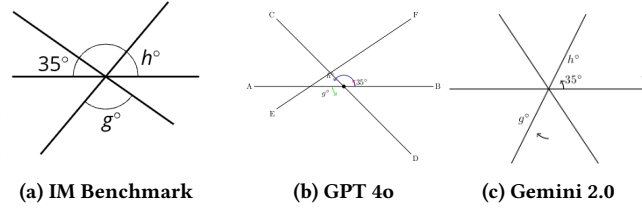


Figure 16: While the generated diagrams (b) and (c) have some visual similarity to the (a) IM Benchmark, they do not adhere to geometric constraints. Both examples do not represent 35° , h° , and the vertical angle of g° as supplementary (adding up to 180°), or g° 's relationship with its vertical angle.

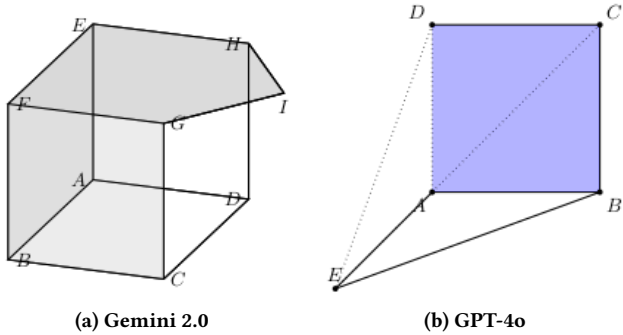


Figure 17: 3D shapes were challenging for multiple reasons. (a) is an example of how models would generate incomplete or poorly formed 3D shapes. (b) is an example of how even though the diagram is well-formed, it misses critical mathematical information important to the question (specifically that it should include a right triangle face). These diagrams were generated with human-enhanced alt-texts and TikZ primer.

5.3.3 Knowledge of Mathematical Diagrams. Models generally had trouble generating several specific diagram types, such as hanger diagrams. In the hanger diagrams in Figure 18, only one condition led to an accurately drawn hanger diagram (z , z , and 2.2 vertically

hung, which conveys that the hanger is balanced), while other conditions typically hung objects separately or were not well-formed. The models seemed to have limited knowledge about what hanger diagrams were and how to draw them, and required explicit instructions from the human-enhanced alt-text. Expressions, Equations, and Inequalities, that utilize balanced hanger diagrams, had the fourth lowest success rate for human-enhanced alt-text at 56.3% (Table 3).

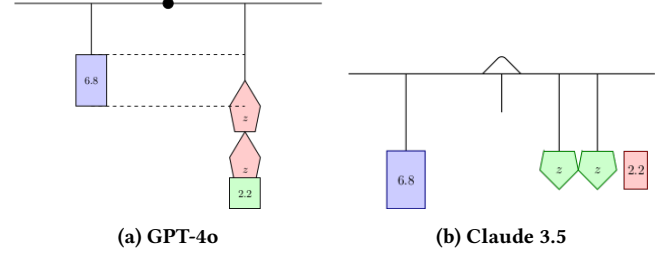


Figure 18: All models struggled with hanger diagrams. Only one condition succeeded (a), while others displayed objects horizontally like in (b), which would actually not be balanced, or were not well-formed at all. It seemed like the models did not know what a hanger diagram was.

5.3.4 Preservation of Mathematical Information and Relationships. The third category of errors involved failure to preserve crucial mathematical information and relationships. In the 3D shape Figure 17, while it does generate a square pyramid, it does not clearly contain a right triangle face that is required from the corresponding math problem.

The tape diagram in Figure 19 is a particularly nuanced example, where the Illustrative Mathematics diagram is complex, containing many proportional relationships essential to the mathematical concept being illustrated. For example, the total length of the top tape diagram C should be longer than Z . Cells with the same label should be the same size. x should be different from y . While all of the diagrams get very close, only 19b captures all of these distinctions. 19c has x and y the same size, and in 19d Z is longer than C . This behavior is interesting given that the human-enhanced alt-text for this diagram included all of these mathematical details.

6 Discussion

Our experimental results demonstrate both the significant potential and current limitations of using Large Language Models for generating mathematical diagrams. The most striking finding is the dramatic impact of high-quality image descriptions. When provided with human-enhanced alt-text, models performed substantially better than with original descriptions, revealing two key insights: first, that the specific language and structure of these descriptions critically impact model performance; and second, that current state-of-the-art models can achieve high success rates when provided with precise, mathematically complete descriptions.

The impact of the TikZ syntax primer varied across models and conditions. While it generally improved compilation rates, its effect on mathematical accuracy was model-dependent. This suggests that

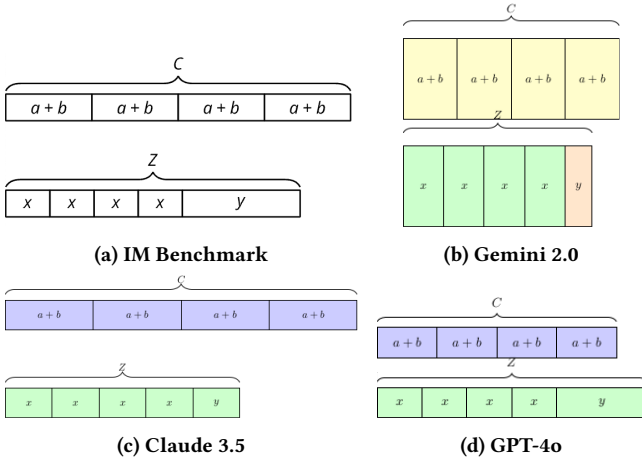


Figure 19: Models had trouble maintaining complex mathematical relationships. Here, only (b) maintains the mathematical distinctions from (a) that are important for solving the problem: that the C tape diagram is longer than the Z one, that x and y are of different lengths. In (c) x and y are the same length, while in (d) C is shorter than Z .

the relationship between technical syntax guidance and mathematical accuracy might be more complex than initially anticipated and may depend on model-specific characteristics.

Our analysis revealed several persistent failure modes that warrant further investigation. These include difficulties with spatial placement, inadequate understanding of specific diagram types’ pedagogical requirements, challenges in maintaining consistency across related diagrams, and limitations in implementing complex geometric constraints. These failures point to fundamental challenges in the current approach of using LLMs to generate mathematical diagrams from natural language descriptions.

These findings have significant implications for both research and practice. For researchers, our results suggest that improving mathematical diagram generation may require specialized architectures that can better handle spatial relationships and geometric constraints, rather than relying solely on enhanced prompting or syntax guidance. For practitioners, while current systems show promise in supporting certain types of diagram generation, their limitations underscore the continued importance of human oversight in educational content creation.

6.1 Limitations and Future Work

Our study has several key limitations that point to promising directions for future research. First, while human-enhanced descriptions significantly improved performance, creating these enhanced descriptions manually is not scalable for the full dataset of 3,793 diagrams. Future work should explore automated methods for improving description quality while maintaining mathematical precision, such as specialized models trained on mathematical language or hybrid approaches combining automated enhancement with efficient human validation.

Second, our evaluation focused only on two binary metrics: compilation and success rates, the latter of which relied on manual review. A more fine-grained evaluation scheme could quantify the key failure modes we identified, as well as alignment with user (i.e. teacher and student) preferences. Such a scheme could facilitate targeted model improvements by helping researchers evaluate and address specific model errors. Further, developing automated diagram evaluations, by extending [14, 27], is critical to enhancing the utility of our dataset as a benchmark. LLM evaluators could take the problem context and description into account when assessing the relevance and accuracy of diagrams.

Third, our current benchmark focuses on English language descriptions. Given the universal nature of mathematical diagrams and their importance in global education, investigating multilingual diagram generation capabilities would be valuable for broader accessibility. Additionally, while Grade 7 provides a rich variety of mathematical diagrams, our findings may not generalize to other grade levels which may feature different types of diagrams.

We see particular promise in better leveraging specialized functions and prior examples. Many mathematical diagrams, such as hanger diagrams or number lines, follow standardized formats with specific constraints. The IM curriculum already includes some custom TikZ functions for these common diagram types, and expanding this library of specialized functions, coupled with clear usage guidance, could significantly improve generation accuracy.

For particularly challenging cases, such as 3D shapes, we hypothesize that few-shot learning approaches or reinforcement fine-tuning could be valuable strategies. We also envision potential benefits from developing self-reviewing agents that can iteratively generate, critique, and refine diagrams [18], as well as interactive systems that could guide users through the diagram specification process through targeted questions. However, this approach is currently limited due to the challenges state-of-the-art LLMs face in understanding and making use of visual information [27], limiting the ability to perform reliable evaluation and quality assurance on mathematical diagrams.

7 Conclusion

This work makes two primary contributions. First, we introduce MathemaTikZ, a comprehensive dataset of mathematical diagrams drawn from the Illustrative Mathematics curriculum, each paired with natural language descriptions and TikZ implementations. This dataset provides a benchmark for evaluating and developing mathematical diagram generation systems, spanning the full range of visualizations used in K-12 mathematics education.

Second, our systematic evaluation of state-of-the-art language models reveals both the significant potential and current limitations of LLM-powered diagram generation. While models can achieve high success rates under optimal conditions, performance varies significantly across diagram types and conditions. Our analysis identifies four key challenges that must be addressed for reliable deployment in educational contexts: spatial reasoning and element placement, adherence to geometric constraints, prevention of mathematical value hallucination, and preservation of mathematical relationships.

Looking ahead, MathemaTikZ provides a foundation for developing more sophisticated approaches to mathematical diagram generation. The dataset offers several valuable use cases for researchers and practitioners. For AI researchers, it serves as a challenging benchmark for evaluating spatial reasoning capabilities in language models. For educational technologists, it can facilitate the development of automated tools that assist teachers in customizing curriculum materials with appropriate visual supports. For curriculum developers, it offers a structured way to analyze and categorize the visual representations that appear throughout K-12 mathematics education.

Beyond evaluation, MathemaTikZ can support fine-tuning specialized models for diagram generation, training systems to convert between different representation formats (e.g., from natural language to TikZ code), and creating interactive educational applications that dynamically generate tailored visualizations based on student needs. The paired nature of the dataset—connecting mathematical problem contexts, descriptions, and implementations—makes it particularly valuable for developing systems that understand the pedagogical intent behind different diagram types.

Our baseline evaluations establish clear metrics for measuring progress in this domain, while our error analysis highlights specific technical challenges that must be addressed. Success in overcoming these challenges could significantly affect how educational content is created and customized, ultimately supporting more effective and accessible mathematics education at scale.

Acknowledgments

This project would not have been possible without the support of Illustrative Mathematics. We are especially grateful to Kristin Um-land, Bill McCallum, and Aurora Ziobrowski for providing access to the IM curriculum data and for their continued support throughout the project. We also thank the Gates Foundation (Grant #068816) and the Stanford Institute for Human-Centered AI for funding this work. Finally, we thank the anonymous reviewers for their helpful feedback.

References

- [1] R. Agodini, B. Harris, S. Atkins-Burnett, S. Heaviside, T. Novak, and R. Murphy. 2009. *Achievement Effects of Four Early Elementary School Math Curricula: Findings from First Graders in 39 Schools (NCEE 2009-4053)*. Technical Report. National Center for Education Evaluation and Regional Assistance, Institute of Education Sciences, U.S. Department of Education.
- [2] Christina A Barbieri, Jessica Rodrigues, Nancy Dyson, and Nancy C Jordan. 2020. Improving fraction understanding in sixth graders with mathematics difficulties: Effects of a number line approach combined with cognitive learning strategies. *Journal of Educational Psychology* 112, 3 (2020), 628.
- [3] Jonas Belouadi, Anne Lauscher, and Steffen Eger. 2024. AutomaTikZ: Text-Guided Synthesis of Scientific Vector Graphics with TikZ. <https://doi.org/10.48550/arXiv.2310.00367> arXiv:2310.00367 [cs]
- [4] D. Blazar, B. Heller, T.J. Kane, M. Polikoff, D. Staiger, S. Carrell, D. Goldhaber, D.N. Harris, R. Hitch, K.L. Holden, and M. Kurlaender. 2019. *Learning by the Book: Comparing Math Achievement Growth by Textbook in Six Common Core States*. Technical Report. Center for Education Policy Research, Harvard University.
- [5] D. Blazar, B. Heller, T.J. Kane, M. Polikoff, D.O. Staiger, S. Carrell, D. Goldhaber, D.N. Harris, R. Hitch, K.L. Holden, and M. Kurlaender. 2020. Curriculum Reform in the Common Core Era: Evaluating Elementary Math Textbooks across Six U.S. States. *Journal of Policy Analysis and Management* 39, 4 (2020), 966–1019. <https://doi.org/10.1002/pam.22257>
- [6] A. J. H. Boonen, F. van Wesel, J. Jolles, and M. van der Schoot. 2014. The Role of Visual Representation Type, Spatial Ability, and Reading Comprehension in Word Problem Solving: An Item-Level Analysis. *Contemporary Educational Psychology* 39, 1 (2014), 59–75. <https://doi.org/10.1016/j.cedpsych.2014.03.005>
- [7] Agneet Chatterjee, Gabriela Ben Melech Stan, Estelle Aflalo, Sayak Paul, Dhruva Ghosh, Tejas Gokhale, Ludwig Schmidt, Hannaneh Hajishirzi, Vasudev Lal, Chitta Baral, and Yezhou Yang. 2024. Getting it Right: Improving Spatial Consistency in Text-to-Image Models. arXiv:2404.01197 [cs.CV] <https://arxiv.org/abs/2404.01197>
- [8] Jennifer L Cooper, Pooja G Sidney, and Martha W Alibali. 2018. Who benefits from diagrams and illustrations in math problems? Ability and attitudes matter. *Applied Cognitive Psychology* 32, 1 (2018), 24–38.
- [9] EdReports. 2021. Kendall Hunt's Illustrative Mathematics (2021). <https://edreports.org/reports/overview/kendall-hunts-illustrative-mathematics-2021>
- [10] Elizabeth A Gunderson, Gerardo Ramirez, Sian L Beilock, and Susan C Levine. 2012. The relation between spatial skill and early number knowledge: the role of the linear number line. *Developmental psychology* 48, 5 (2012), 1229.
- [11] Mary Hegarty and Maria Kozhevnikov. 1999. Types of visual-spatial representations and mathematical problem solving. *Journal of educational psychology* 91, 4 (1999), 684.
- [12] Andrew P. Jaciw, Wendy M. Hegseth, Li Lin, Melissa Toby, Denis Newman, Boya Ma, and Jenna Zacamy. 2016. Assessing Impacts of Math in Focus, a "Singapore Math" Program. *Journal of Research on Educational Effectiveness* 9, 4 (2016), 473–502. <https://doi.org/10.1080/19345747.2016.1164777>
- [13] Rijul Jain, Wode Ni, and Joshua Sunshine. 2023. Generating Domain-Specific Programs for Diagram Authoring with Large Language Models. In *Companion Proceedings of the 2023 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*. ACM, Cascais Portugal, 70–71. <https://doi.org/10.1145/3618305.3623612>
- [14] Mehran Kazemi, Hamidreza Alviri, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. 2023. GeomVerse: A Systematic Evaluation of Large Models for Geometric Reasoning. arXiv:2312.12241 [cs.CV] <https://arxiv.org/abs/2312.12241>
- [15] Jaewook Lee, Jeongah Lee, Wanyong Feng, and Andrew Lan. 2025. From Text to Visuals: Using LLMs to Generate Math Diagrams with Vector Graphics. arXiv:2503.07429 [cs.AI] <https://arxiv.org/abs/2503.07429>
- [16] Jeongwoo Lee, Kwangsuk Park, and Jiyeon Park. 2024. VISTA: Visual Integrated System for Tailored Automation in Math Problem Generation Using LLM. <https://doi.org/10.48550/arXiv.2411.05423> arXiv:2411.05423 [cs]
- [17] Rizwaan Malik, Dorna Abdi, Rose Wang, and Dorottya Demszy. [n. d.]. Scaffolding Middle-School Mathematics Curricula With Large Language Models. ([n. d.]). <https://doi.org/10.26300/B47Y-MH41>
- [18] S. Mondal, X. Li, and K. Peterson. 2024. SciDoc2Diagrammer-MAF: Towards Generation of Scientific Diagrams from Documents guided by Multi-Aspect Feedback Refinement. To appear in EMNLP (Findings).
- [19] Aki Murata. 2008. Mathematics teaching and learning as a mediating process: The case of tape diagrams. *Mathematical Thinking and Learning* 10, 4 (2008), 374–406.
- [20] Vitali Petsiuk, Alexander E. Siemenn, Saisamrit Surbehera, Zad Chin, Keith Tyser, Gregory Hunter, Arvind Raghavan, Yann Hicke, Bryan A. Plummer, Ori Kerret, Tonio Buonassisi, Kate Saenko, Armando Solar-Lezama, and Iddo Drori. 2022. Human Evaluation of Text-to-Image Models on a Multi-Task Benchmark. arXiv:2211.12112 [cs.CV] <https://arxiv.org/abs/2211.12112>
- [21] Palak Roy, Ruth Staunton, and Helen Poet. [n. d.]. ChatGPT in Lesson Preparation - A Teacher Choices Trial. <https://doi.org/10.1186/ISRCTN13420346>
- [22] Johanna Schoenherr, Anselm R Strohmaier, and Stanislaw Schukajlow. 2024. Learning with visualizations helps: A meta-analysis of visualization interventions in mathematics education. *Educational Research Review* (2024), 100639.
- [23] J. Sun, N. Kim, and S. Ju. 2024. AI-Driven Feedback for Enhancing Students' Mathematical Problem-Solving: The ScaffoldiaMyMaths System. In Proceedings of the International Conference on Computers in Education (ICCE).
- [24] Elizabeth Warren and Tom J Cooper. 2009. Developing mathematics understanding and abstraction: The case of equivalence in the elementary years. *Mathematics Education Research Journal* 21, 2 (2009), 76–95.
- [25] Katherine Ye, Wode Ni, Max Krieger, Dor Ma'ayan, Jenna Wise, Jonathan Aldrich, Joshua Sunshine, and Keenan Crane. 2020. Penrose: From Mathematical Notation to Beautiful Diagrams. *ACM Transactions on Graphics* 39, 4 (Aug. 2020). <https://doi.org/10.1145/3386569.3392375>
- [26] E. Zala, D. Freedman, and Y. Liu. 2023. DiagrammerGPT: Generating Open-Domain, Open-Platform Diagrams via LLM Planning.
- [27] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. 2024. MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? <https://doi.org/10.48550/arXiv.2403.14624> arXiv:2403.14624 [cs]
- [28] Qilong Zhangli, Jindong Jiang, Di Liu, Licheng Yu, Xiaoliang Dai, Ankit Ramchandani, Guan Pang, Dimitris N. Metaxas, and Praveen Krishnan. 2024. Layout Agnostic Scene Text Image Synthesis with Diffusion Models. arXiv:2406.01062 [cs.CV] <https://arxiv.org/abs/2406.01062>