MathVizAI: An AI-Powered System for Context-Aware Mathematical Visualization from Textbooks

Saianuj Sambarapu[1], Naroju Siddhartha[2], Dharavath Praveen[3], Dr S. Shyam Sunder Rao[4], Battula Balnarsaiah[5]

[1,2,3,5]Student, Computer Science and Engineering (AI & ML), Nalla Malla Reddy Engineering college, Hyderabad, India
[4]Professor, AI&ML, Nalla Malla Reddy Engineering College

Abstract
Mathematical concepts presented in textbooks are predominantly symbolic and static, which often limits learners' ability to develop strong conceptual understanding. Although animated visualizations can improve comprehension, creating such content manually is time-consuming and requires specialized expertise. This paper presents MathVizAI, an AI-powered system that automatically converts mathematical content from textbooks into short, context-aware animated visualizations. The proposed system integrates mathematical optical character recognition, symbolic parsing, large language model–based semantic rea- soning, and programmatic animation generation using Manim. Given a selected region from a textbook, MathVizAI extracts mathematical expressions, interprets their conceptual structure, generates a structured animation plan, and renders an explana- tory video without human intervention. The system emphasizes semantic consistency, visual clarity, and structured explanation rather than direct symbolic transcription. Experimental results on undergraduate-level mathematics content demonstrate that the pipeline can reliably generate executable animations for a majority of extracted expressions, supporting the feasibility of automated mathematical visualization for educational use.

Keywords: Mathematical OCR, LaTeX extraction, large language models, programmatic animation, Manim, automated mathematical visualization

Introduction

Mathematics education relies heavily on textbooks as the primary medium for formal instruction. While mathemati- cally rigorous, textbooks predominantly present concepts using static symbols, equations, and diagrams, which often fail to convey dynamic relationships or procedural intuition. Prior studies in mathematics education indicate that learners benefit significantly from visual and animated representations, espe- cially when dealing with abstract topics such as calculus, linear algebra, and geometry [1], [2].

Recent advances in artificial intelligence have introduced new opportunities for augmenting educational content. Large Language Models (LLMs) have demonstrated strong capabil- ities in generating mathematical explanations, diagrams, and structured reasoning [3], [4]. Parallel work has explored the use of AI for generating educational videos and microlearning content, showing promise in learner engagement and motiva- tion [5], [6]. However, most existing systems either rely on manually curated inputs or operate at the level of short-form summaries rather than directly processing textbook material.

A critical technical challenge lies in extracting mathemati- cal content from textbooks. Mathematical Optical Character Recognition (OCR) remains difficult due to complex nota- tion, layout variability, and symbol ambiguity. Recent neural OCR systems such as MixTeX and other document parsing frameworks have improved LaTeX extraction accuracy, yet they primarily focus on transcription rather than semantic interpretation or visualization [7]–[9]. As a result, the extracted equations remain static representations, offering limited peda- gogical value on their own.

Separately, animation and visualization tools such as Manim have proven effective for explaining mathematical ideas through programmatic animations, as popularized in STEM education [10], [11]. Interactive platforms like GeoGebra support spatial reasoning but require manual construction and expert intervention [2]. More recent agent-based systems, including Code2Video and Preacher, automate video generation from code or documents but are not specialized for math- ematical textbooks and often lack contextual mathematical narration [12], [13].

Motivated by these gaps, this paper presents MathVizAI, an AI-powered system that automates the transformation of mathematical content from textbooks into context-aware animated visualizations. The system integrates mathematical OCR, large language models, and the Manim animation engine to generate short explanatory videos directly from textbook inputs. Unlike prior approaches, MathVizAI emphasizes end- to-end automation, contextual storytelling, and error-aware processing tailored specifically to mathematical content.

This work focuses on system design and implementation rather than large-scale educational evaluation. The contribu- tions of this paper are threefold: (1) an integrated pipeline for converting textbook mathematics into animated visual explanations, (2) an error-aware architecture that addresses OCR noise and LLM-generated code failures, and (3) an open, extensible framework suitable for educators and self-learners.

Related Work

Research related to MathVizAI spans three primary ar- eas: AI-assisted mathematics education, mathematical content extraction from documents, and automated visualization and video generation.

A. AI in Mathematics Education

Recent advances in large language models have enabled new forms of AI-assisted learning in mathematics. Studies have explored the ability of LLMs to generate explanations, solve mathematical problems, and support extended curricula [3]. While these models demonstrate strong linguistic and sym- bolic reasoning capabilities, their outputs are primarily text- based and lack visual grounding. Empirical studies comparing AI-generated and human-created educational videos indicate that visual presentation plays a significant role in learner engagement and motivation [1]. Systematic reviews of ge- ometry education further emphasize the importance of spatial visualization tools in improving conceptual understanding [2]. However, most AI-based tutoring systems focus on adaptive feedback or content generation rather than automated visual- ization of textbook material.

B. Mathematical OCR and Document Parsing

Extracting mathematical expressions from textbooks re- mains a challenging problem due to complex notation, two- dimensional layouts, and symbol ambiguity. Neural OCR systems such as MixTeX have demonstrated improved perfor- mance in LaTeX extraction while maintaining efficient CPU inference [7]. Additional work in handwritten mathematical expression recognition introduces attention-based architectures and data augmentation techniques to improve robustness [15]. More recent document parsing frameworks, including Mon- keyOCR, incorporate structural and relational modeling to bet- ter interpret scientific documents [9]. Despite these advances, existing OCR systems primarily aim at transcription accuracy and do not address downstream semantic interpretation or visualization of extracted expressions.

C. Visualization and Educational Video Generation

Programmatic animation tools such as Manim have been widely adopted for explaining mathematical concepts through dynamic visualizations [10], [11]. While effective, these tools require manual scripting and domain expertise, limiting scala- bility. Interactive platforms like GeoGebra support exploratory learning but rely on user-driven construction [2]. Recent AI- driven approaches attempt to automate video generation from code or documents. Systems such as Code2Video and Preacher generate educational videos using agent-based pipelines [12], [13], while VideoAgent focuses on personalized scientific video synthesis [14]. Other works explore generating diagrams and vector graphics directly from text using LLMs [4]. Al- though promising, these systems are not specifically designed for mathematical textbooks and often lack error-aware han- dling of symbolic content.

D. Positioning of MathVizAI

MathVizAI builds upon these lines of work by integrating mathematical OCR, large language models, and programmatic animation into a single end-to-end pipeline. Unlike prior ap- proaches, it directly processes textbook inputs and emphasizes contextual mathematical storytelling through animation. The system focuses on practical integration challenges, such as OCR noise, LLM-generated code failures, and rendering con- straints, positioning MathVizAI as a system-level contribution rather than a benchmark-driven model.

Proposed Method

This section describes the proposed MathVizAI method for transforming static mathematical content from textbooks into short, pedagogically structured animated visualizations. The method does not rely on training new models; instead, it integrates existing OCR, large language models, and program- matic animation tools within a controlled and fault-tolerant pipeline. The emphasis is on structured intermediate repre- sentations, validation-driven generation, and visual clarity. An overview of the proposed method is shown in Fig. 1.

A. Method Overview

Given a textbook page in PDF or image format, Math- VizAI converts selected mathematical content into a concise explanatory animation. The method is organized as a sequence of five stages: (1) input acquisition and region selection, (2) mathematical content extraction, (3) semantic structuring and explanation planning, (4) programmatic animation generation, and (5) validated video rendering. Each stage produces an ex- plicit intermediate output, enabling modularity, error isolation, and extensibility.

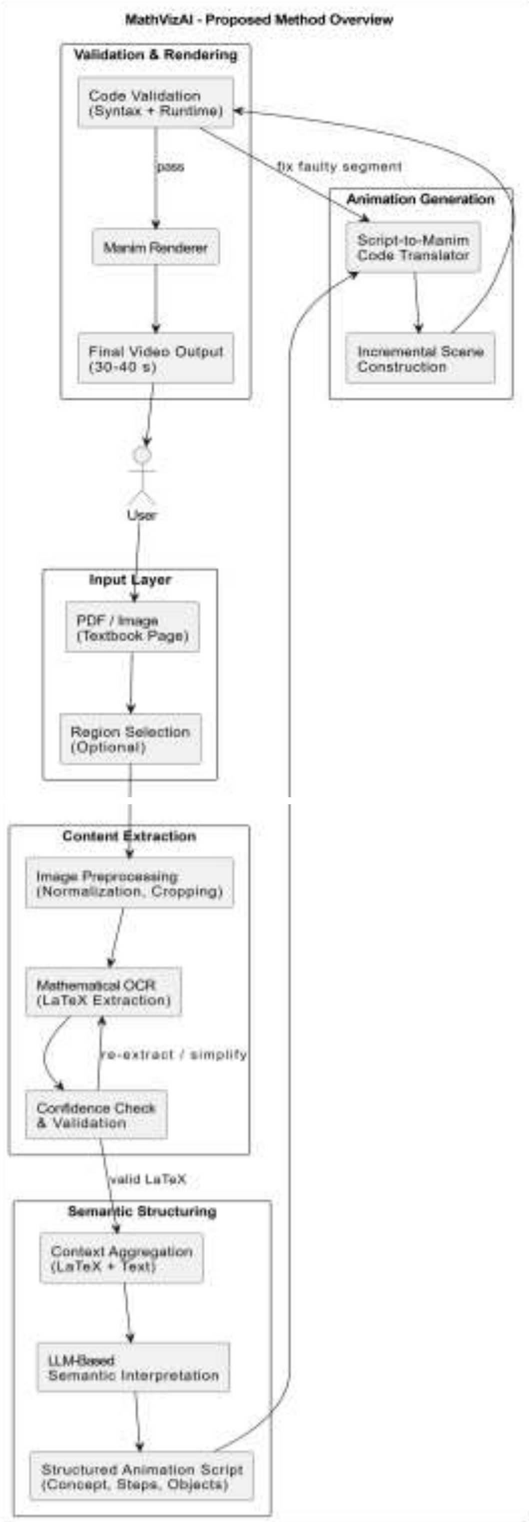

Fig. 1. Overview of the MathVizAI method for transforming textbook mathematics into animated explanatory videos

B. Input Acquisition and Mathematical Content Extraction

The input consists of scanned textbook pages or digital PDFs. Users may optionally select a region of interest cor- responding to a specific equation, derivation, or explanation. The selected region is preprocessed through normalization, denoising, and cropping to reduce background artifacts.

Mathematical optical character recognition is then applied to extract symbolic expressions and convert them into structured latex representations. Since mathematical OCR is sensitive to layout complexity and symbol ambiguity, the extracted output is accompanied by a confidence score. When confidence falls below a predefined threshold, the system avoids generating a fully detailed animation and instead triggers simplified visualization or re-extraction, preventing the propagation of incorrect symbolic information.

C. Semantic Structuring and Explanation Planning

Raw latex expressions are insufficient for direct visualiza- tion, as they do not encode conceptual flow or explanatory intent. To address this, the extracted Latex expressions are combined with surrounding textual context and passed to a large language model for semantic interpretation.

Rather than producing free-form explanations, the model generates a structured animation script that explicitly defines:

- the underlying mathematical concept,
- key variables and symbols,
- a step-wise decomposition of the explanation,
- visual objects to be introduced at each step,
- temporal ordering and transition constraints.

Fig. 2. Internal processing stages of the MathVizAI system

This structured script acts as an intermediate control layer between symbolic mathematics and animation code, as il- lustrated in Fig. 2. By constraining the output format, the method reduces logical inconsistencies, limits visual clutter, and enables systematic control over explanation depth and difficulty level.

D. Programmatic Animation Generation

The structured animation script is translated into executable animation code using the Manim framework. Each script step is mapped to elementary visual primitives such as axes, curves, geometric objects, and mathematical annotations. Temporal sequencing is explicitly defined to emphasize mathematical progression rather than static presentation.

To mitigate errors introduced during code generation, the method avoids monolithic scene construction. Instead, an- imation components are generated incrementally, allowing individual segments to be validated and corrected without regenerating the entire scene.

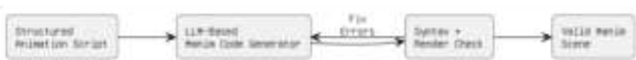

Fig. 3. Script-driven animation generation with iterative validation and correction

E. Iterative Validation and Video Rendering

Generated animation code undergoes both syntactic and execution-level validation prior to rendering.

Syntax checks ensure valid imports and object definitions, while execution checks verify scene construction and runtime stability. If a failure occurs, only the faulty segment is regenerated under constrained prompts, forming an iterative validation loop as shown in Fig. 3.

Once validated, the animation is rendered into a short video clip, typically 30–40 seconds in duration. Rendering param- eters are selected to prioritize visual clarity and conceptual alignment over photorealism. The final output is stored and returned as a standalone video suitable for integration into digital learning environments and self-study workflows.

Experimental Setup

A. Test Corpus

To evaluate the practical behavior of MathVizAI, we con- structed a test corpus consisting of 42 textbook-style mathe- matical expressions spanning multiple domains. The corpus in- cludes simple arithmetic expressions, algebraic equations, cal- culus problems (derivatives and integrals), matrix operations, trigonometric expressions, and complex composite formulas. The distribution of test cases across categories is shown in Fig. 4.

The selected expressions were sourced from undergraduate- level mathematics textbooks and lecture materials, ensuring realistic layout structures and notation complexity representa- tive of commonly encountered educational content.

B. Execution Environment

All experiments were conducted on a local workstation run- ning a 64-bit operating system. The MathVizAI pipeline was executed end-to-end without manual intervention, including OCR-based expression extraction, semantic interpretation via a large language model, Manim code generation, and video rendering.

Rendering was performed using the default Manim renderer with standard resolution and frame rate settings to ensure consistency across test cases and to avoid hardware-specific optimization.

C. Evaluation Protocol

Each mathematical expression in the corpus was processed independently through the full MathVizAI pipeline. For every input, the system attempted to generate a complete animation video without user correction. In cases where rendering er- rors occurred, the system applied its constrained regeneration mechanism to repair localized code failures rather than regen- erating the entire animation.

For each test case, we logged intermediate and final pipeline outcomes, including OCR detection success, script generation status, render success, regeneration attempts, and total execu- tion time. Metrics were computed over repeated runs using identical inputs to verify stability and reproducibility.

D. Evaluation Metrics

Since MathVizAI is a system integration framework rather than a predictive model, evaluation focuses on pipeline ro- bustness, execution reliability, and efficiency. We report the following metrics:

- **Expression Coverage Rate (ECR):** the proportion of detected mathematical expressions that result in a valid visualization.
- **Script-to-Render Success Rate (SRR):** the fraction of generated animation scripts that render successfully.
- **First-Pass Success Rate (FPSR):** the percentage of scripts that render correctly without requiring regener- ation.
- **Regeneration Recovery Rate (RRR):** the proportion of initially failed renders that are

successfully recovered through constrained regeneration.

- **Semantic Preservation Proxy (SPP):** a rule-based check verifying consistency between extracted mathematical symbols and rendered visual primitives.

RESULTS AND DISCUSSION

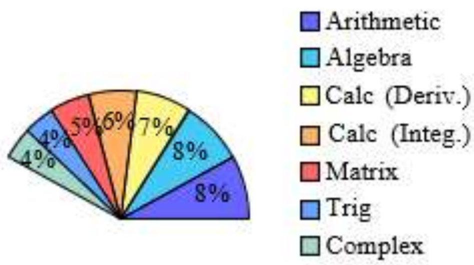

Fig. 4. Distribution of 42 test cases across mathematical categories used in evaluation.

A. Quantitative Results

Quantitative evaluation was conducted by executing Math- VizAI on textbook-style mathematical inputs and logging outcomes at each stage of the pipeline. Table II summarizes pipeline-level proxy metrics that characterize robustness and execution reliability.

The Expression Coverage Rate (ECR) of 66.7% indicates that approximately two-thirds of OCR-detected mathemati- cal expressions were successfully converted into rendered visualizations. Failures at this stage were primarily caused by malformed latex outputs and expressions involving un-supported symbolic constructs, highlighting the sensitivity of downstream stages to OCR quality.

The Script-to-Render Success Rate (SRR) of 75.0% demon- strates that most LLM-generated Manim scripts were syntac- tically and semantically valid. A First-Pass Render Success Rate (FPSR) of 68.8% further suggests that the majority of scripts executed successfully without requiring corrective intervention.

The Regeneration Recovery Rate (RRR) of 20.0% indicates limited recovery of initially failed renders through constrained regeneration. While this confirms the feasibility of localized correction, it also exposes challenges in precise error localiza- tion and semantic repair.

Finally, the Semantic Preservation Proxy (SPP) score of 8.3% reflects the difficulty of enforcing strict symbol-level correspondence between extracted mathematical expressions and rendered visual elements. This low value is partially at- tributable to intentional visual abstraction and the conservative nature of the proxy metric. As a result, the SPP score should be interpreted as a strict lower bound on symbolic alignment rather than a measure of conceptual correctness.

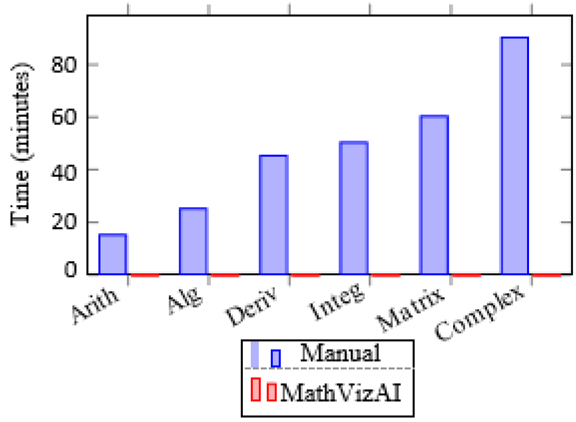

Fig. 5. Comparison of animation generation time between manual Manim scripting and MathVizAI across expression types.

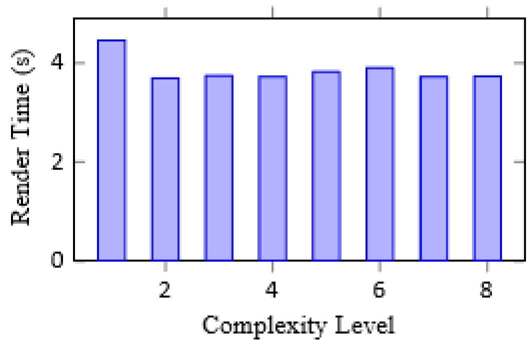

Fig. 6. Average render time across increasing equation complexity levels, demonstrating stable performance.

(Here, complexity level denotes an ordinal measure of symbolic and visual difficulty rather than algorithmic complexity. Levels are defined based on the number of mathematical symbols, operators, visual primitives, and animation steps required to construct the scene.)

B. Pipeline-Level Proxy Metrics

Since MathVizAI is designed as a system integration frame- work rather than a predictive model, evaluation focuses on pipeline robustness and execution reliability. We report proxy metrics that characterize how effectively mathematical inputs are transformed into rendered animations.

Expression Coverage Rate: Expression Coverage Rate (ECR) measures the proportion of detected mathematical ex- pressions that result in a valid visualization:

$$ECR = \frac{N_{\text{visualized}}}{N_{\text{detected}}}$$

where N_{detected} denotes the number of equations extracted by the OCR stage and $N_{\text{visualized}}$ represents the subset successfully translated into animation scripts. Failures were most com- monly associated with malformed LATEX outputs or expressions requiring unsupported symbolic constructs.

Script-to-Render Success Rate: Execution reliability is evaluated using the Script-to-Render Success Rate (SRR),

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

TABLE I
DEFINITION OF EQUATION COMPLEXITY LEVELS

Level	Description
1–2	Simple arithmetic, single operator expressions
3–4	Algebraic equations with multiple symbols
5–6	Calculus expressions and matrix operations
7–8	Composite expressions combining symbolic and geometric elements

defined as:

$$SRR = \frac{N_{\text{successful renders}}}{N_{\text{generated scripts}}}$$

This metric captures the system's ability to generate syntacti- cally and semantically valid Manim code. While some scripts fail during the initial render attempt, constrained regeneration improves final render success by correcting localized errors without regenerating entire scenes.

Regeneration Efficiency: MathVizAI employs iterative correction rather than full regeneration when rendering errors occur. Regeneration efficiency is assessed by observing the proportion of failed scripts that are successfully rendered after one or more targeted correction cycles. This behavior reflects the stabilizing effect of structured intermediate representations in the animation pipeline.

Semantic Preservation Check: A rule-based semantic preservation check ensures consistency between extracted mathematical expressions and rendered visuals. This check verifies the presence of core variables, equation structure, and corresponding visual primitives in the final animation. While this does not measure pedagogical effectiveness, it provides a deterministic indication of symbolic-to-visual fidelity.

TABLE II
PIPELINE-LEVEL EVALUATION RESULTS FOR MATHVIZAI

Metric	Value
Expression Coverage Rate (ECR)	66.7%
Script-to-Render Success Rate (SRR)	75.0%
First-Pass Render Success Rate (FPSR)	68.8%
Regeneration Recovery Rate (RRR)	20.0%
Semantic Preservation Proxy (SPP)	8.3%

LIMITATIONS AND DISCUSSION

This work presents a system-level evaluation of MathVizAI, and several factors may influence the interpretation and gen- eralizability of the reported results.

Input Selection Bias

The evaluation was conducted on a limited set of textbook- style mathematical inputs primarily drawn from algebra, co-ordinate geometry, and introductory calculus. While these domains are representative of commonly taught undergraduate material, they do not capture the full diversity of mathematical notation, such as advanced proofs, multi-page derivations, or highly symbolic areas like abstract algebra. Consequently, the reported metrics may not generalize uniformly across all mathematical subfields

Dependence on OCR Quality

Pipeline performance is strongly influenced by the accuracy of mathematical OCR. Errors introduced during LaTeX ex- traction propagate downstream and directly affect expression coverage and semantic preservation metrics. Although con- temporary OCR models were employed, mathematical OCR remains an open research challenge, particularly for dense layouts, nested expressions, and multi-line equations. Improve- ments in OCR accuracy would likely yield proportional gains across subsequent pipeline stages.

Proxy Nature of Evaluation Metrics

Several reported metrics, including Expression Coverage Rate and Semantic Preservation Proxy, serve as proxy mea- sures rather than direct indicators of educational effective- ness. These metrics capture execution reliability and symbolic consistency but do not directly assess learner comprehension or instructional value. The Semantic Preservation Proxy, in particular, enforces strict symbol-level alignment and may underestimate conceptual correctness in cases where deliberate visual abstraction is applied for explanatory clarity.

Limited Regeneration Analysis

The regeneration strategy employed in MathVizAI focuses on localized correction of rendering failures. While this ap- proach improves execution stability and avoids full scene re- generation, the current evaluation does not explore alternative regeneration policies or deeper semantic repair mechanisms. As a result, the reported regeneration recovery rate reflects the present system configuration rather than an upper bound on recoverable failures.

Absence of Human-Centered Evaluation

This study does not include user studies or human eval- uation of learning outcomes. The absence of learner-facing experiments limits claims regarding pedagogical effectiveness. However, this design choice was intentional, as the primary contribution of this work lies in system integration and automated visualization rather than instructional assessment. Future work will incorporate human-centered evaluations to complement the pipeline-level analysis.

Overall, these limitations do not invalidate the reported findings but instead clarify the scope of conclusions that can be drawn. The evaluation is intended to characterize system robustness and integration behavior rather than to establish pedagogical superiority.

CONCLUSION AND FUTURE WORK

This paper presented MathVizAI, an end-to-end system for automatically transforming mathematical textbook content into short, context-aware animated visualizations. Rather than introducing a new predictive model, MathVizAI emphasizes system integration, combining mathematical OCR, large language models, and programmatic animation to reduce the manual effort required to produce high-quality educational visuals.

Experimental evaluation using pipeline-level proxy met- rics shows that MathVizAI can reliably generate executable animations for a substantial portion of detected mathemat- ical expressions. Expression Coverage and Script-to-Render success rates indicate that the system performs effectively on common algebraic, geometric, and introductory calculus content, while iterative regeneration improves robustness in the presence of LLM-generated code errors. Qualitative results further suggest that temporally structured visual narration helps reveal mathematical relationships that are difficult to convey through static textbook representations.

At the same time, the evaluation highlights several limita- tions. Errors introduced during OCR propagate through down- stream stages, and semantic preservation remains challenging when mathematical context is implicit or densely encoded in textbook layouts. In addition, the system prioritizes visual clarity over symbolic completeness, which can lead to the abstraction of intermediate derivation steps. These limitations reflect deliberate design trade-offs aimed at producing short, interpretable animations rather than exhaustive formal proofs. Future work will focus on improving semantic fidelity through tighter coupling between symbolic reasoning and visual primitives, incorporating confidence-aware OCR fusion, and extending support to more advanced mathematical topics. Additional directions include user-in-the-loop correction mechanisms, adaptive explanation depth based on learner profiles, and large-scale human evaluation studies to better assess educational impact. Together, these extensions aim to advance MathVizAI toward a more reliable and pedagogically grounded framework for automated mathematical visualization.

References

1. T. Netland et al., “Comparing human-made and AI-generated teaching videos: Effects on learning and motivation,” *Computers & Education*, vol. 205, p. 105164, 2024, doi:10.1016/j.compedu.2024.105164.
2. R. Marasabessy et al., “Fostering spatial visualization in GeoGebra- assisted geometry lessons: A systematic review,” *EURASIA Journal of Mathematics, Science and Technology Education*, vol. 20, no. 12, 2024.
3. M. Alsayyad and F. Kadhem, “Can the current trends of AI handle a full course of mathematics?,” arXiv:2507.21664, 2025.
4. J. Lee, J. Lee, W. Feng, and A. Lan, “From text to visuals: Using LLMs to generate math diagrams with vector graphics,” arXiv:2503.07429, 2025.
5. M. Constantinides et al., “The reel deal: Designing and evaluating LLM- generated short-form educational videos,” arXiv:2509.05962, 2025.
6. F. Sadique et al., “Next-gen education: Enhancing AI for microlearning,” arXiv:2508.11704, 2025.
7. R. Q. Luo and Y. Xu, “Mixtex: Multimodal LATEX OCR with CPU inference,” arXiv:2406.17148, 2024.
8. H. He, M. Ye, J. Zhang, X. Cai, J. Liu, B. Du, and D. Tao, “Reasoning- OCR: Can large multimodal models solve complex reasoning problems from visual cues?,” arXiv:2505.12766, 2025.
9. Z. Li, Y. Liu, Q. Liu, Z. Ma, Z. Zhang, S. Zhang, Z. Guo, J. Zhang, X. Wang, and X. Bai, “MonkeyOCR: Document parsing with a structure- recognition-relation triplet paradigm,” arXiv:2506.05218, 2025.
10. The Manim Community Developers, “Manim – Mathematical animation framework,” 2025. [Online]. Available: https://www.manim.community/
11. C. Zhang, “Manim for STEM education: Visualizing complex problems through animation,” arXiv:2510.01187, 2025.
12. J. Liu, L. Yang, H. Luo, F. Wang, H. Li, and M. Wang, “Preacher: Paper-to-video agentic system,” arXiv:2508.09632, 2025.
13. Y. Chen, K. Q. Lin, and M. Z. Shou, “Code2video: A code-centric paradigm for educational video generation,” arXiv:2510.01174, 2025.
14. X. Liang et al., “Videoagent: Personalized synthesis of scientific videos,” arXiv:2509.11253, 2025.
15. Z. Li, L. Jin, S. Lai, and Y. Zhu, “Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention,” arXiv:2007.10092, 2020.
16. M. W. F. Ku, T. Chong, J. Leung, K. Shah, A. Yu, and W. Chen, “Theoremexplainagent: Towards video-based multimodal explanations for LLM theorem understanding,” arXiv:2502.19400, 2025.
17. W. Wu et al., “Mind’s eye of LLMs: Visualization-of-thought elicits spatial reasoning in large

language models," arXiv:2404.03622, 2024.
18. L. Blecher, G. Cucurull, T. Scialom, and R. Stojnic, "Nougat: Neu- ral optical understanding for academic documents," arXiv:2308.13418, 2023.
19. C. T. G. Lau, H. E. Tang, and T. M. Chien, "Exploring foundation students' AI math tool preferences, duration of use, and confidence in mathematics," *Computers & Education*, 2024