

Augmented Math: ML-Based Approach to Authoring AR Explorable Explanations by Augmenting Static Math Textbooks

Neil Chulpongsatorn*

University of Calgary
Calgary, Canada

thobthai.chulpongsat@ucalgary.ca

Mille Skovhus Lunding*

Aarhus University
Aarhus, Denmark

University of Calgary
Calgary, Canada
milledsk@cs.au.dk

Nishan Soni

University of Calgary
Calgary, Canada
nishan.soni@ucalgary.ca

Ryo Suzuki

University of Calgary
Calgary, Canada
ryo.suzuki@ucalgary.ca

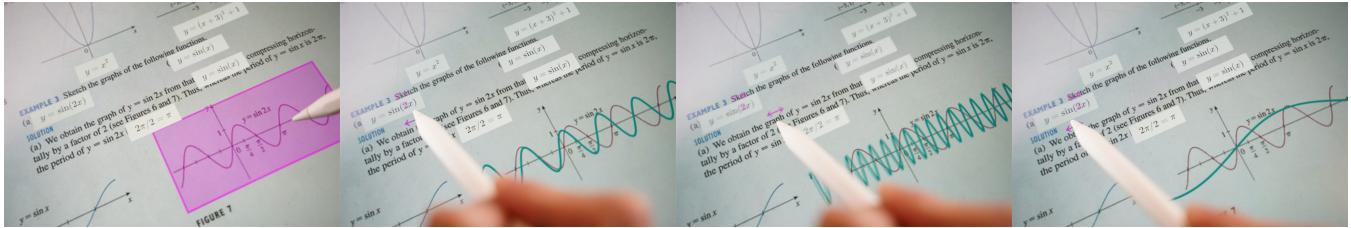


Figure 1: Augmented Math is an authoring tool that transforms static math textbooks into on-demand AR-based explorable explanations. The figure illustrates how a user can use our system to explore changes in the sine curve graph by binding and manipulating the extracted equation in the mobile AR interface.

ABSTRACT

We introduce Augmented Math, a machine learning-based approach to authoring AR explorable explanations by augmenting static math textbooks without programming. To augment a static document, our system first extracts mathematical formulas and figures from a given document using optical character recognition (OCR) and computer vision. By binding and manipulating these extracted contents, the user can see the interactive animation overlaid onto the document through mobile AR interfaces. This empowers non-technical users, such as teachers or students, to transform existing math textbooks and handouts into on-demand and personalized explorable explanations. To design our system, we first analyzed existing explorable math explanations to identify common design strategies. Based on the findings, we developed a set of augmentation techniques that can be automatically generated based on the extracted

content, which are 1) dynamic values, 2) interactive figures, 3) relationship highlights, 4) concrete examples, and 5) step-by-step hints. To evaluate our system, we conduct two user studies: preliminary user testing and expert interviews. The study results confirm that our system allows more engaging experiences for learning math concepts.

CCS CONCEPTS

• Human-centered computing → Mixed / augmented reality.

KEYWORDS

Augmented Reality; Explorable Explanations; Interactive Paper; Augmented Textbook; Authoring Interfaces

ACM Reference Format:

Neil Chulpongsatorn, Mille Skovhus Lunding, Nishan Soni, and Ryo Suzuki. 2023. Augmented Math: ML-Based Approach to Authoring AR Explorable Explanations by Augmenting Static Math Textbooks. In *UIST '23: The ACM Symposium on User Interface Software and Technology, Oct 23–28, 2023, San Francisco, CA, U.S.A.* ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/123456.123456>

*Both authors contributed equally to the paper

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 ACM 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.

UIST '23, Oct 23–28, 2023, San Francisco, CA, U.S.A.

© 2023 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/123456.123456>

1 INTRODUCTION

“People currently think of text as information to be consumed. I want text to be used as an environment to think in.” — Bret Victor [62]

Today’s textbooks, whether in digital or physical format, primarily

consist of *static explanations* that only allow users to passively consume information, without facilitating a deep understanding that comes from interactive dialogue and exploration. The concept of *explorable explanations* [62] has been developed to shift this paradigm, transforming text into an interactive medium to think in, rather than just a source of information to be consumed. Such dynamic and interactive explanations have enormous potential as learning aids, as they foster deeper understanding of abstract concepts through engaging and playful explorations, which is otherwise difficult to achieve through static textbooks alone. Therefore, explorable explanations have emerged as a promising approach to changing the way we learn complex concepts across various educational domains, such as math [17], physics [46], programming [19, 59, 64], data science [13], and machine learning [15, 53, 65].

However, creating these explorable explanations often requires a significant amount of time and effort, as well as substantial programming expertise. This prevents non-technical users, such as teachers or students, from creating their own interactive content tailored to their specific needs and context. As a result, they are often limited to passively utilizing existing interactive content available on the internet, while the majority of their own textbooks remain static.

In this paper, we propose Augmented Math, a new machine learning (ML) enabled approach to creating explorable explanations by *augmenting* static math textbooks, rather than programming them from scratch. Our system first scans a static document to extract values, symbols, and graphs using optical character recognition (OCR) and computer vision techniques. The system then localizes the position of detected math formulas and graphs, converting them into the computer understandable mathematical expression. The user can then author interactive content by binding and manipulating these extracted elements. These interactive visual outputs are embedded and overlaid on the scanned document either through mobile AR for printed paper or through a desktop interface for scanned PDFs. This empowers non-technical end users to transform their own textbooks into on-demand, personalized, explorable explanations without requiring programming knowledge.

To design our system, we first analyze the common design strategies widely used in the existing interactive math explanations available on various websites. We identify three high-level categories of these strategies: 1) exemplify through concrete values, 2) visualize through interactive and animated graphs, and 3) guide through contextual hints or exercise. Based on these design strategies, we develop a set of augmentation techniques for our proof-of-concept system, including 1) dynamic values, 2) interactive figures, 3) relationship highlights, 4) concrete examples, and 5) step-by-step hints. These features are designed to be automatically generated based on the extracted content, while covering a wide variety of strategies identified through our analysis.

To evaluate our system, we conduct two user studies. First, we perform an exploratory study with eleven participants, comparing our system across three conditions: 1) static paper, 2) web-based interface, and 3) mobile AR. The study results confirm that the AR-based interface is the most engaging, while the web interface provides a better system usability score (SUS). Qualitative feedback informs us that the system helps them understand mathematical concepts through interactive and playful exploration, compared

to static textbooks. In the second study, we conducted expert interviews with five math instructors to quantitatively compare our approach to existing learning resources and practices, such as hand-outs, videos, and existing interactive websites. In addition, we investigate how our tool could fit their potential needs and practices, seeking the opportunity for real-world use scenarios. Their feedback suggests that our approach has great potential as an on-demand and personalized learning assistant tailored for their specific context, which is not well-supported with the currently available learning practices. Based on their feedback and insights, we discuss how to expand our proposed approach beyond the current proof-of-concept prototype for future deployment.

Finally, this paper contributes to the followings:

- (1) Augmented Math, a novel machine learning-based approach to creating explorable explanations by augmenting static math textbooks.
- (2) A set of features that consist of our augmented math textbooks, informed by a taxonomy analysis of existing practices.
- (3) Results and insights from preliminary user testing and expert interviews.

2 RELATED WORK

2.1 Explorable Explanations

Explorable explanations [62] have emerged as an increasingly popular practice to teach and learn abstract concepts across various educational contexts, such as math, science, and engineering (e.g. *PhET* [46], *Distill* [15]). Interactive explanations that leverage animations, simulations, and gaming provide more engaging experiences, naturally encouraging students to develop their understanding of complex concepts in a playful manner [1, 23, 28, 65]. Many of these math augmentation techniques were demonstrated in Bret Victor’s KillMath projects [63]. Various tools, such as *Idyll* [12], *Flapjax* [41], *GeoGebra* [17], *Mavo* [61], and *Data Theater* [33], have been similarly developed to lower the barrier to create interactive explanations. For example, *Idyll Studio* [14] allows users to create data-driven explanations and *GeoGebra* [17] lets users make interactive diagrams to teach various math topics such as geometry, algebra, and calculus.

However, existing tools still require the programming expertise that makes it difficult for non-technical users to adapt and create their own interactive content. Head et al. [22] explicitly mention that the development of an appropriate authoring tool still requires future work, and there is a strong need for intelligent design assistants that automatically augment static math textbooks. Towards this goal, *CrossData* [10] supports non-technical users in creating data-driven documents by binding text and data in the authoring process. This allows for easy updating of text, data, and graphs, but the output document is static once the authoring process is finished. Beyond textual explanations, some tools support non-technical users in creating interactive diagrams for math and engineering education. For example, *MathPad2* [34] allows users to create math sketches that turn into interactive diagrams. In *PhysicsBook* [8] and *PhysInk* [52], students can create interactive sketches that animate through given math formulas or physics simulation. Alternatively, *Kitty* [27] explores an authoring interface to let users make interactive animation by binding between different components, which

is useful for engineering education. In these tools, however, the interactive explanations are based on sketches that are disconnected from existing material such as textbooks.

In contrast, this paper proposes to *repurpose* existing static documents to turn them into explorable explanations. This approach allows users to easily and quickly create interactive content with minimal time and effort as it does not require programming or generating content from scratch. Additionally, to the best of our knowledge, our work is the first to investigate AR-based explorable explanations, which also have great potential for enhancing the learning experience by providing immersive and interactive content in a real-world context. This approach opens up new possibilities for engaging and personalized learning, bridging the gap between physical textbooks and digital interactivity.

2.2 Augmented Paper

Since Wellner’s first demonstration of *DigitalDesk* [66], HCI researchers have actively explored the idea of *augmented paper* by overlaying digital content onto physical documents [21]. Researchers have investigated various ways to augment physical paper, including projecting digital information (e.g., *AffinityLens* [56], *Mouse-Light* [54], *Qook* [67]), illuminate physical paper (e.g., *IllumiPaper* [31]), and embedding content in mobile AR (e.g., *Opportunistic Interfaces* [16], *Teachable Reality* [42]) or see-through displays (e.g. *MagicBook* [5], *Mixed Reality Book* [18], *HoloDoc* [39], *Replicate and Reuse* [20]). However, most of these works assume prepared content, and only a few works have looked into the *authoring aspect* of these augmented documents. For example, *PapARVis Designer* [9] allows users to author embedded visualizations for static paper, and *Dually Noted* [48] enables users to annotate physical documents by leveraging computer vision to recognize the layout of documents. Most closely related to our work, *Paper Trail* [49] explores the authoring of augmented paper based on simple user-defined animation. Our work builds on top of these works to enable AR-based documents with a greater focus on interactive exploration compared to prior works. By emphasizing explorability, we aim to create more engaging and personalized learning experiences through augmented paper.

2.3 AR for Math Education

Augmented reality has shown advantages in teaching math by enhancing visualizations [3] and creating playful learning experiences [11, 29]. Such AR-based tools explore various math topics such as geometry [26, 51, 57] and algebra [11]. For example, *AR-Math*[25] demonstrates teaching early math skills such as counting and basic geometry by turning everyday objects into mathematical learning materials, using AR and computer vision. Similarly, *RealitySketch*[58] also aims to visualize math and physics concepts in the real world by embedding responsive graphics through an AR sketching interface. However, in most of these existing works, the AR contents are disconnected from existing learning materials like textbooks. A few studies have focused on augmenting existing math textbooks. For example, Li et al. [38] turn a math book into an interactive game to attract children’s interest, however the connection between the physical book and AR content is limited to multiple-choice questions. *GeoAR* [30] creates an augmented book

to teach geometry, with AR geometric shapes appearing on the pages, but the AR content is predefined and only works with the specific book. In contrast, our work introduces a general-purpose authoring tool that enables teachers and students to create their own augmentations using existing formulas and figures in math textbooks.

2.4 AR Authoring Tools

Existing AR authoring tools often rely on user-sketched content (e.g. *Rapido* [36], *Pronto* [37], *RealitySketch* [58], *Sketched Reality* [24], *ProjectAR* [40]), user-created physical models (e.g. *ProtoAR* [43]), or pre-defined virtual models (e.g. *Reality Composer* [4], *Adobe Aero* [2]). However, there is a lack of authoring tools that enables the users to augment existing visual content such as physical paper. Moreover, creating dynamic AR content typically requires user-defined demonstrations (e.g. *Rapido* [36], *Pronto* [37]) or pre-defined animations (e.g. *Reality Composer* [4], *Adobe Aero* [2]). While these approaches work for general-purpose AR prototyping, they are not well-suited for AR-based explorable explanations, where animation should be bound to mathematical formulas or simulations. To address these limitations, our proposed authoring tool leverages machine learning and data-binding techniques. This allows users to extract existing content from documents to create interactive AR explanations tailored to their needs, facilitating the creation of engaging educational materials.

3 DESIGN STRATEGIES OF EXISTING EXPLORABLE MATH EXPLANATIONS

3.1 Design Strategy Analysis

To design our authoring interface, we first try to understand the common techniques and design strategies employed in existing explorable explanations.

3.1.1 Motivation. Although previous work, such as *Augmented Math* [22], have presented taxonomy analyses in similar domains, their focus is broader, encompassing the readability of math formulas in books, papers, slides, and videos. While these analyses provide insights into low-level augmentation and annotation techniques, such as visual notations or styles, we need to understand higher-level design strategies to inform our system features. Therefore, our analysis specifically focuses on *interactive websites* to better understand the high-level *design strategies* employed in these websites. Our analysis complements existing taxonomy analyses presented in [22] to gain a deeper understanding of explorable math explanations.

3.1.2 Dataset. We collected 43 existing interactive math explanations available on the internet from various sources. These datasets were gathered from notable collections of interactive websites, such as *Explorabl.es* [6], *Gallery of Concept Visualization* [45], *GeoGebra* [17], *PhET* [44], *Explained Visually* [47], *Visualize It* [7], *Interactive Maths* [50], *Seeing Theory* [32], *Distill* [15], and *Awesome Interactive Math Tools* [60], as well as through the individual searches. A complete list of the collected dataset with each website link is available in Appendix A.

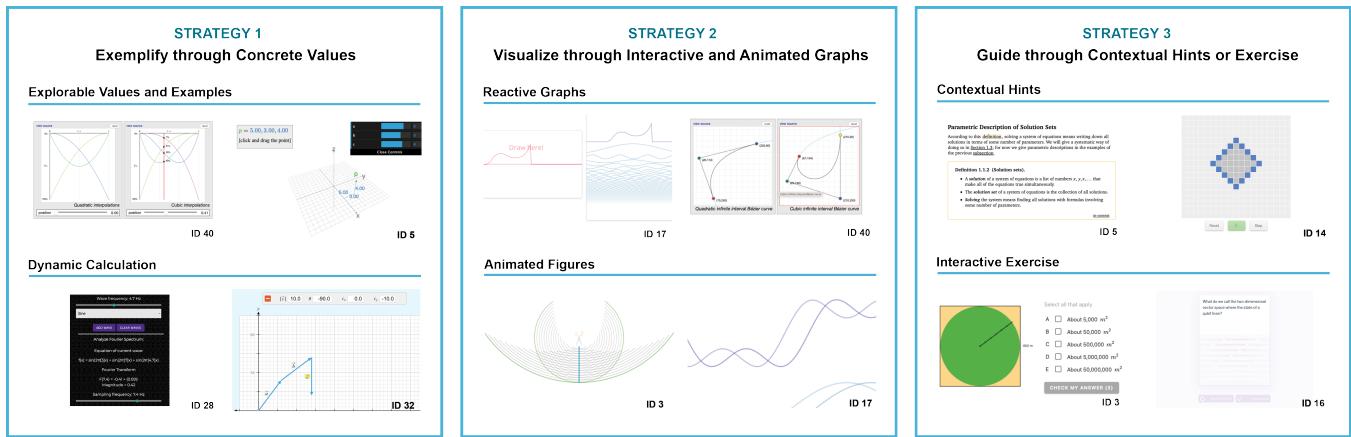


Figure 2: Design strategy analysis based on existing interactive math explanations

3.1.3 Method. We identified and categorized high-level design strategies through an open-coding analysis. After collecting the examples, two of the authors first conducted an open-coding analysis to identify initial categories and dimensions. Next, all authors reflected on the initial categorization to discuss consistency and comprehensiveness. After refining the categories, three authors performed systematic coding with individual tagging to categorize the complete dataset. Finally, the authors reflected on individual tagging to resolve discrepancies and obtain the final coding results.

3.1.4 Results. Our analysis led to the identification of three high-level design strategies: 1) Exemplify through concrete values, 2) Visualize through interactive and animated graphs, and 3) Guide through contextual hints and exercises. For each strategy, we also identify two common design components, as illustrated in Figure 2. These collected design strategies span a wide range of math topics, including *Algebra* (6/43), *Geometry* (9/43), *Calculus* (2/43), *Probability* (11/43), *Arithmetic* (1/43), *Applied math* (19/43), and *Graph theory* (3/43). In addition to the high-level design strategies, we also identify other low-level design and interaction elements, which include *Slider and Scrubbable* (27/43), *Direct Graph Manipulation* (18/43), *Options* (3/43), *Text Input* (5/43), *Scroll* (2/43), *Button* (24/43), and *Hover* (3/43). The results are summarized in Appendix A (Figure 13 as well as Table 1 and Table 2). In the following sections, we discuss each design strategy and its examples in more detail. Figure 2 also illustrates the visual summary of each strategy and design component.

Strategy 1. Exemplify through Concrete Values

Explorable Values and Examples. The first strategy is to exemplify abstract and symbolic math representations with concrete values. Such examples include providing concrete examples for abstract concepts, embedding numerical values in each symbol of complex math formulas, and showing concrete probabilities with simulation experiments. Most of the websites allow users to dynamically update these values with interactive components such as sliders or text inputs.

Dynamic Calculation. Another important aspect of this strategy is that when users change these values, then corresponding equations also change based on dynamic calculations and simulations. This allows students to immediately see how each value affects the outcome, helping them gain an intuition of how symbols and equations are related to each other. By leveraging exploratory examples and dynamic calculations, students can develop a better understanding of abstract concepts by visualizing them.

Strategy 2. Visualize through Interactive and Animated Graphs

Reactive Graphs. While the first strategy mainly focuses on textual representation, another common strategy leverages dynamic visual representation. In particular, existing websites often use reactive graphs that can be dynamically changed based on corresponding interactions or value changes. Such examples include a reactive graph of a trigonometric function (algebra), an interactive diagram for the Pythagorean theorem (geometry), and an exploratory explanation of Newton's method (calculus). This technique is often used in conjunction with the exemplifying strategy described above. For example, when a user changes a value in an equation, the graphs and diagrams are also updated. This helps users understand the relationship between the symbolic variable and its visual representation.

Animated Figures. Alternatively, animated figures focus primarily on animation with less emphasis on interactivity. Animated figures are particularly useful for visualizing simulation results. In this way, animation can visualize how simulation evolves with temporal representation. Examples include showing simulation results of probability, demonstrating various geometric transformations, and temporal visualization of sine and cosine curves. While this representation focuses more on animation than interactivity, users can still manipulate parameters such as timelines.

Strategy 3. Guide through Contextual Hints or Exercises

Contextual Hints. Contextual hints provide additional information to students when they need it. Examples include referencing definitions to remind the context or breaking down complex solutions through step-by-step instructions. This support can be customized based on a student's progress and can adapt to their individual learning needs, ensuring that they receive the most appropriate help at the right time.

Interactive Exercises. Incorporating exercises directly into the explanations allows users to actively engage with the content. Exercises can come in various forms, such as multiple-choice questions for a simple quiz, fill-in-the-blank problems to solve equations, or direct manipulation of figures to answer questions. Providing immediate feedback on these exercises can help students identify their mistakes and reinforce their understanding of the concepts.

4 AUGMENTED MATH: SYSTEM AND DESIGN

4.1 Overview

This section introduces Augmented Math, an authoring tool for AR-based explorable explanations. The system aims to empower non-technical users, such as students and instructors, to create interactive explanations without programming expertise. To achieve this, our system augments static math textbooks by embedding and overlaying interactive content. The authoring workflow comprises the following steps: 1) *scan*: users first scan a math textbook or handout with a camera, 2) *extract*: the system automatically extracts math formulas and graphs shown in the document using OCR and computer vision, 3) *select*: users select extracted content like symbols, values, equations, and graphs to make them interactive, 4) *bind*: users bind selected elements together, 5) *manipulate*: users manipulate the values and data of selected elements by dragging, and 6) *update*: the system propagates changes responsively updates based on data binding and manipulation. The system supports both mobile AR and desktop interfaces for making printed paper or scanned PDFs interactive, respectively.

4.2 Scope

We largely focus our investigations around high-school level mathematics based on the United States standard curriculum. Specifically, the primary topics are: algebra, trigonometry, geometry, and calculus. Augmented Math does not have features such as complex illustration nor abstract equations to support higher level topics such as statistics probability and quantum-mechanics.

4.3 Example Authoring Walkthrough

In this section, we describe an example authoring and interaction walkthrough using our system. We use an example from a high-school calculus textbook (Single Variable Calculus [55] P.39) that presents an equation, $y = x^2 + 6x + 10 = (x + 3)^2 + 1$, and its corresponding graph. The purpose of this augmentation is to visualize how values of a , b , c , and n affect the graph of $y = (x - a)^n + b$. We describe how users can transform this static content into an explorable explanation using our system.

Step 1: Scan and Extract Math Textbook or Handout. The first step is to scan a printed document or PDF from an existing math textbook. After the user takes a photo to scan it, the system automatically extracts text, math formulas, and graphs using OCR and computer vision. In Figure 3-1, math formulas captured by our system are shown as extracted content. Our system converts math equations to computer-readable LaTeX expressions like $y = (x + 3)^2 + 1$, enabling semantic understanding of the given math formula. OCR extracts written text and math formulas, while custom computer vision algorithms extract figures and graphs, allowing users to select and draw a graph overlaid on the existing figure.

Step 2: Select Equations and Bind them to Graphs. After scanning and extracting the document, users can author augmented content via mobile AR or desktop interfaces. To make the textbook interactive, users first select extracted values, symbols, equations, and graphs. When a user selects a graph, the system highlights the selection as illustrated in Figure 3-1. The system automatically extracts the x-axis, y-axis, origin, and drawn graph line using computer vision. When the user also selects the equation $y = (x+3)^2+1$, the system automatically binds and draws a graph based on the selected equation (Figure 3-2). This AR-embedded graph enables dynamic manipulation and updating based on user interaction. Users can also bind multiple equations to a single graph to visually compare equations by overlaying them.

Step 3: Drag to Change the Variable Values. Users can dynamically change values by dragging symbols or variables. For example, in Figure 3-4, when the user drags the 3 in the equation $y = (x + 3)^2 + 1$, the system automatically replaces the value of 3, treating it as a dynamic variable. In Figure 3-7, users can specify values for variables like x and y , displaying horizontal and vertical lines related to the current x and y values on the graph. Changing a value alters the corresponding variable and equation based on dynamic calculation.

Step 4: Enable Bi-Directional Binding between Variable and Graph. Changing a value prompts the system to update the graph dynamically. Figure 3-5 and -8 show how users can modify the graph plot of $(x + a)^n + b$ by dragging the graph or the variable values of b and n , respectively. For example, adjusting the a value of $(x + a)^2 + b$ shifts the graph horizontally, while changing the b moves it vertically. Bi-directional binding allows users to drag the graph, which in turn changes the corresponding variable value. This reactive feature facilitates user interaction with numerical values, graphs, and charts. By adjusting variables in a formula, users can observe their effects on the final output, allowing them to develop an intuition of abstract relationships, which are otherwise hard to understand.

4.4 Supported Augmentation Features

To facilitate diverse interactive explorations for high-school level mathematics, we have developed the following five augmentation techniques: 1) *dynamic values*, 2) *interactive figures*, 3) *relationship highlights*, 4) *concrete examples*, and 5) *step-by-step hints*. These features are designed to be automatically generated based on the

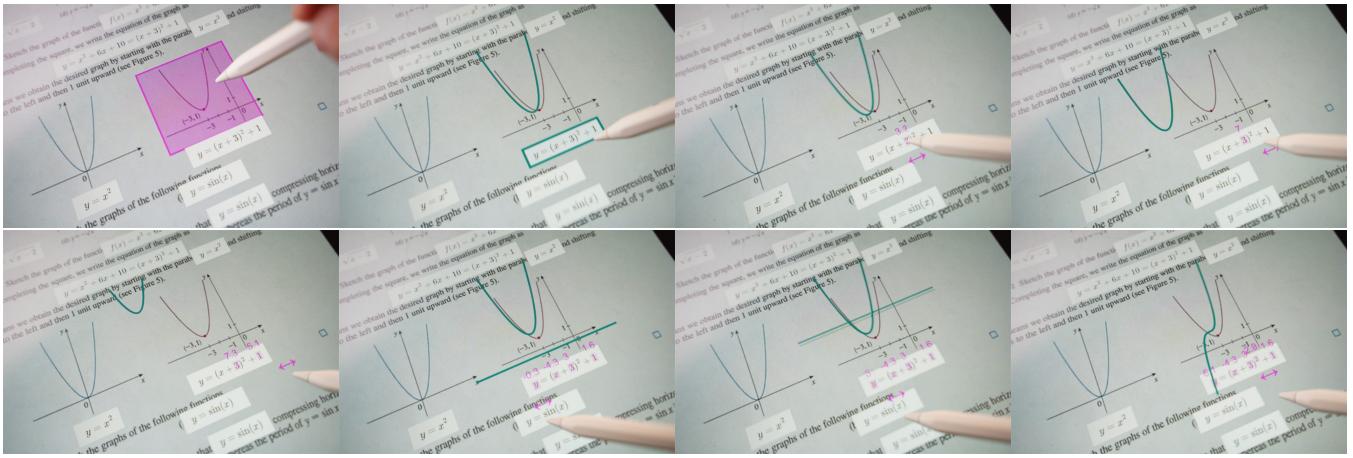


Figure 3: An example authoring and interaction walkthrough to make an explorable explanation for a quadratic equation and its graph. From top left: 1) The system first extracts math equations and graphs from a scanned document and lets the user select an extracted figure, 2) When the user selects an equation to bind with the selected graph, the system automatically draws a graph given the equation, 3-4) As the user drags one of the symbols, the value, equation, and corresponding graphs update, 5-8) The user can also manipulate other symbols and elements, reflecting changes in values and graphs for interactive explorations.

extracted content, with the goal of encompassing a wide range of strategies identified through analysis.

4.4.1 Dynamic Values. Dynamic values allow users to insert and manipulate concrete values for extracted symbols and variables found in the document. This feature is inspired by “*strategy 1: exemplify through concrete values*” from our taxonomy analysis. For instance, Figure 4 demonstrates users setting and adjusting values of h , k , and r in the equation $\sqrt{(x - h)^2 + (y - k)^2} = r^2$.

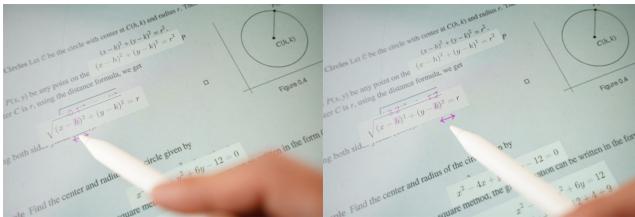


Figure 4: Dynamic Values

Since the system processes mathematical equations as LaTeX expressions rather than plain text, it can semantically comprehend and dynamically compute equations based on the inserted values. Additionally, the system enables users to manipulate these values across various formulas throughout the document, generating responsive explanations that demonstrate how specific changes impact multiple equations through dynamic calculations. Dynamic values can be integrated with other features, such as interactive figures, which we discuss next.

4.4.2 Interactive Figures. Interactive figures offer users the opportunity to comprehend concepts through explorable visual representations. Unlike static images, these interactive figures promote

intuitive understanding by allowing users to interact with them. This feature is inspired by “*strategy 2: visualize through interactive and animated graphs*” from our taxonomy analysis.

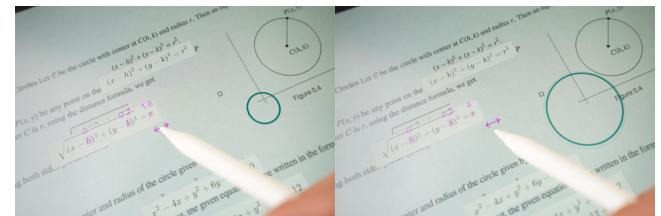


Figure 5: Interactive Figures

Users can make a graph interactive by selecting and binding equations to it, as described in the previous section. For instance, Figure 5 demonstrates how changing the value of r dynamically affects the circle’s radius in the graph accordingly.

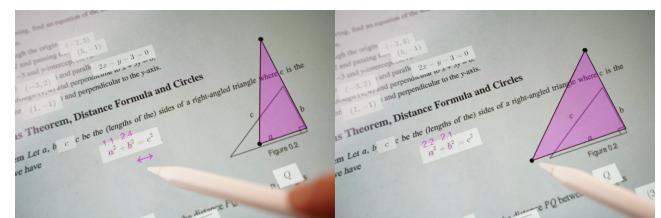


Figure 6: Relationship Highlights

Interactive figures also enhance the understanding of geometric concepts. For example, Figure 6 shows how changing a variable in the equation $a^2 + b^2 = c^2$ updates the corresponding triangle shape,

enabling users to grasp the geometric concept more effectively. These interactive figures support bi-directional binding, allowing users to drag the graph or shape, which in turn updates the values accordingly.

4.4.3 Relationship Highlights. Relationship highlights enable users to visualize connections between different variables and visual references. This feature is inspired by “*strategy 2: visualize through interactive and animated graphs*” from our taxonomy analysis. Relationship highlights can explicitly reveal the connection between variables and visual references, facilitating intuitive comprehension for subjects like geometry or calculus.

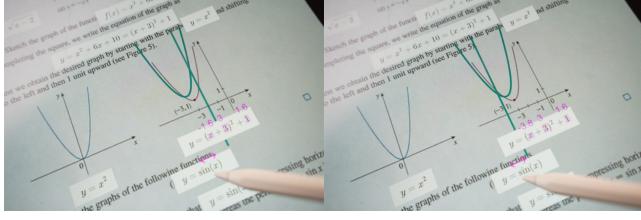


Figure 7: Relationship Highlights

For example, Figure 7 shows the relationship highlights in the graph by holding the pen over x to show the current values on the graph. Similarly, when users hold over a in the triangle example (Figure 6), the corresponding line is highlighted, showing which variable represents the associated visual reference in the figure and graphs.

4.4.4 Concrete Examples. Concrete examples aid users by illustrating abstract concepts through specific instances. This feature is inspired by both “*strategy 1: exemplify through concrete values*” and “*strategy 3: guide through contextual hints or exercises*” from our taxonomy analysis. By identifying operations such as summation, the system can present concrete examples to contextualize and break down abstract formulas.

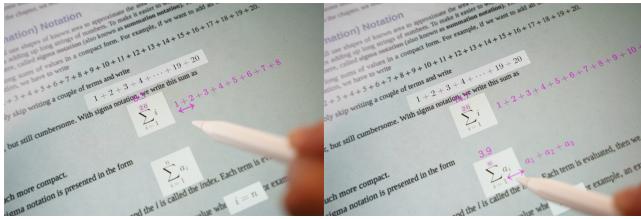


Figure 8: Concrete Examples

For instance, Figure 8 illustrates the user dragging the values of 20 and n in both $\sum_{i=1}^{20} i$ and $\sum_{i=1}^n a_i$. The system then exemplifies the formula by showing $1 + 2 + \dots + n$ or $a_1 + a_2 + \dots + a_n$. Concrete examples provide users with explorable instances to enhance their understanding of abstract concepts.

4.4.5 Step-by-Step Hints. Step-by-step hints offer contextual assistance by breaking down complex math problems or equations into a series of step-by-step instructions and solutions. This feature

is inspired by “*strategy 3: guide through contextual hints or exercises*” from our taxonomy analysis. The system automatically computes the breakdown of a formula by performing arithmetic operations, simplifications, and decompositions.

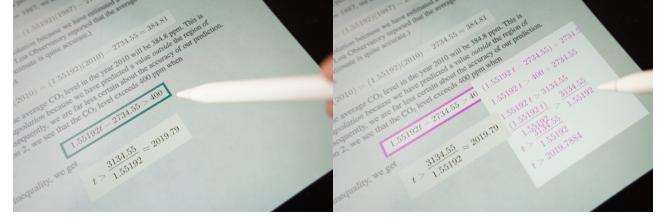


Figure 9: Step-by-Step Hints

For example, Figure 9 demonstrates the user selecting an inequality $1.55192t - 2734.55 > 400$ and then breaking it down into a series of step-by-step hints. Step-by-step hints are applicable to various equations and problems, such as solving quadratic equations like $x^2 - 7x + 10 = 0 \Rightarrow (x - 5)(x - 2) = 0 \Rightarrow x = 2, 5$. By decomposing and breaking down the equation through automated step-by-step instructions, users receive contextual feedback that enhances their understanding.

4.5 Implementation

Our system is implemented using WebAR interface that leverages A-Frame, 8th Wall, and HTML Canvas. For graph manipulation and symbolic computation, we employ the MathJS library. For text and math formula extraction, we utilize Google Cloud, MathPix, and CnSTD OCR. Finally, we develop a custom computer vision algorithm based on OpenCV for figure and graph extraction. To enhance reproducibility, we make our source code publicly available as open-source¹ and a desktop interface is accessible through a live demo link².

4.5.1 Math Formula Extraction. To extract text and math formulas from the input document, we employ OCR techniques. We first apply Google Cloud OCR to detect all text and localize the position of each letter and word. Google Cloud OCR outputs plain text without appropriate math expressions, making it difficult to understand the mathematical content semantically. Therefore, we implement other techniques to convert the scanned document to the LaTeX expression with MathPix OCR³. Since MathPix does not provide location information for each detected formula, we also employ CnSTD⁴, an OCR toolkit based on PyTorch and MXNet, that can detect the accurate position of inline and independent line math formulas within the document. CnSTD math formula detection algorithm is trained with CnMFD Dataset⁵ that contains various math formulas across 17,500 pages of documents. Upon localizing the math in the document, we compare the similarity between detected plain text and math formulas based on the position of the localized math

¹Anonymized for review

²<https://ryosuzuki.github.io/math/>. Note that the AR version requires a paid 8th Wall API subscription; therefore we only provide a web-based interface as a live demo

³<https://mathpix.com/>

⁴<https://github.com/breezedeus/CnSTD>

⁵https://github.com/breezedeus/CnMFD_Dataset

equation, allowing us to accurately localize the position of each math expression and overlay it on the printed document.

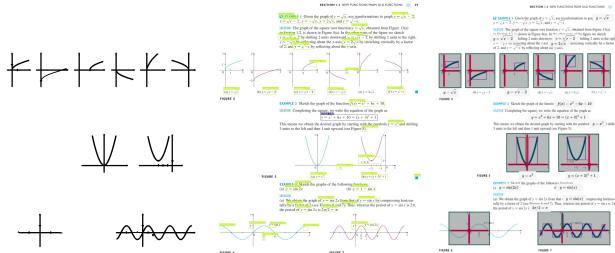


Figure 10: Example results of figure extraction, math formula extraction, and graph identification.

4.5.2 Figure and Graph Extraction. Since OCR only enables the detection and extraction of textual information, we developed a custom computer vision algorithm to accurately localize the positions of graphs and diagrams. To achieve this, we first remove text and extract only diagrams from the given document by using OpenCV contour line extraction. By setting the minimum length of the contour line length, we filter out text and extract only graphs and diagrams. Once the figures are extracted, the system processes the contour line data to detect bounding boxes, x-axes, y-axes, origins, and graph paths. To identify appropriate x-y coordinates, we extract the longest horizontal and vertical lines in the given figure and use them as the x-axis and y-axis. We then use graph paths to identify the appropriate ratio for a given pixel value.

4.5.3 Authoring Interface and Document Tracking. Our authoring interface is implemented using A-Frame, a WebAR framework based on JavaScript and HTML. Visual elements are rendered through an HTML Canvas element embedded as a dynamic texture of a plane object in the A-Frame scene. We use the Konva library for Canvas manipulation. The detected mathematical expressions are rendered using SVG with the MathJax library. Graphs and diagrams are dynamically rendered through a series of lines, with points computed using the MathJS⁶ symbolic computation engine. We also employ the MathJS library for step-by-step hints and concrete examples features. The overall interface is developed using JavaScript and the React.js framework. For document tracking, we utilize 8th Wall's image tracking features⁷, which provide more reliable tracking than alternatives like MindAR and AR.js.

5 TECHNICAL EVALUATION

To evaluate the accuracy of our pipeline, we collected 14 different textbooks which include 5 Algebra, 4 Trigonometry, 2 Geometry, and 3 Calculus textbooks. We selected 10 random pages from each of the textbooks for our sample. We performed our detection pipeline to a total of 140 pages of the collected samples and then check the results manually (as there is no standardized data set or method). We measured the error rate for **textual** equation recognition (Math-Pix, CnSTD, Google OCR), as well as **visual** graph and geometry detection (OpenCV and Custom Detection).

⁶<https://mathjs.org/>

⁷<https://www.8thwall.com/>

5.1 Results

The results suggest that the math equation recognition works fairly well with 75% correctness. These are often very minor errors such as and extra spaces and an extra newline. Among 140 pages, there are 63 pages with graphs and 20 pages with geometries, which makes up 59% of all the samples. Out of these 83 image samples, graph detection was accurate 48 out of 63 times (76%), while geometry detection was only accurate 8 out of 20 times (40%). By investigating these results, we also noticed some error patterns. For graph detection, we noticed that decoration lines or tables are detected as false positives and multiple X-Y scale lines were detected as the X-Y axis. In addition, for geometry detection, the auxiliary lines often introduce the difficulty of the appropriate detection. In general, a visually cluttered textbook with a lot of colors, shapes, or visuals often generates many errors. Based on this technical evaluation, we also expand the discussion about the current limitations and failure examples.

6 USER STUDY

To evaluate the potential of Augmented Math, we conducted a user study consisting of preliminary user testing with eleven students and expert interviews with six math instructors. The purpose of the user testing was to evaluate the ease of use and user satisfaction with the system, while the expert interviews aimed to gain insights for educational use from the experts' perspectives.

6.1 Preliminary User Evaluation

We evaluated the usability of our system in three different conditions: the Augmented Math mobile AR interface, the Augmented Math desktop interface, and static textbook pages without AR. The same pages from a high school math textbook were used for each condition. We recruited 11 adult participants, all of whom had at least a high school level knowledge in math and varying experiences with AR applications. 6 of the participants were Man, 1 was a woman, while 5 people preferred not to disclose this information. There were 5 people whose age ranges from 25-29, 1 person who is between 35-39, 1 person who is below 20, and 5 that preferred not to say. Each session lasted approximately 45 minutes, and participants were compensated with 10 USD for their time.

6.1.1 Method. The study employed a within-subject evaluation design, with the three conditions randomized across participants. The AR interface was displayed on a tablet mounted on a tripod, with a pen used for interaction, while the web version ran on a laptop, with a mouse for interaction. The study began with an introduction that included participants completing a consent form, followed by a system walkthrough. Subsequently, participants were asked to answer five math questions in each of the conditions, targeting the different features of the system:

- How would the graph change if a particular constant were to be increased or decreased?
- How would the equation change if a particular constant were to be increased or decreased?
- How would you simplify this equation?
- How would you expand this notation?

- How would the variables in this equation change if we were to change the shape it represents in certain ways?

The graphs, equations, and numbers varied slightly between the conditions. After completing all conditions, participants filled out a questionnaire that included usability questions for the AR and desktop interfaces, inquiries about each feature of the system, and an overall comparison of the three conditions. Responses were recorded on a 7-point Likert scale, and open questions allowed participants to provide additional feedback.

6.1.2 Results. This section describes the results of our preliminary user study. We describe three aspects: 1) system usability between our mobile AR and desktop interfaces, 2) overall experience and engagement to compare mobile AR, desktop, and static textbooks, and 3) feedback for different features.

Usability of Mobile AR and Desktop Interface. Overall, participants found both the AR interface and desktop interface easy to use, with a preference for the desktop interface. The desktop interface received an average SUS score of 81.82 ($SD = 13.56$), while the AR interface had an average SUS score of 75 ($SD = 15.32$). Many participants encountered issues with the imprecision of the AR version, including small fonts on the page that made selections difficult, especially when viewed through the camera (P1-2, P5, P9-11). This problem was exacerbated by the shakiness of participants' hands (P5) and the device itself when in use (P2, P10). In contrast, participants found the web version smoother to interact with, as the mouse provided a more precise input modality (P1, P2) and the fields were more consistently identified by the OCR (P3). However, some participants desired even greater precision, such as through text box input (P10, P11).

Comparison of the Three Conditions: Mobile AR, Desktop, and Static Textbook. When comparing the three conditions, the desktop was the favorite among participants ($M = 6.00$, $SD = 1.26$), followed by the AR version ($M = 5.18$, $SD = 0.98$). The static textbook pages without AR were the least preferred ($M = 3.40$, $SD = 1.81$). When asked about which condition was the most engaging, participants preferred the AR interface ($M = 6.09$, $SD = 0.83$) over the desktop interface ($M = 5.82$, $SD = 1.07$) due to its novelty (P7, P10) and the intuitive physical manipulation it allowed (P11). The static textbook scored an average of 3.9 ($SD = 1.97$) in engagement. The results are summarized in Figure 11.

Comparison of the Different Features. Participants generally provided positive feedback about the features. On a scale of 1 to 7, the *interactive figure* received an average score of 6.55 ($SD = 0.69$) and was particularly popular among participants. They appreciated being able to see the cause and effect immediately on the graph as they interacted with the equation. The second favorite feature was the *relationship highlight*, which scored an average of 5.81 ($SD = 1.6$). Participants valued how it helped them see the correlation between the shape and the equation more clearly. However, one participant (P2) did not find this feature helpful, as they believed the shape was already too abstract and disconnected from the equation. Most participants also found the *step-by-step hint* helpful ($M = 5.81$, $SD = 1.6$), particularly as a convenience (P4, P7) or as a means of checking for mistakes (P9, P10). One participant (P2), however, found that it

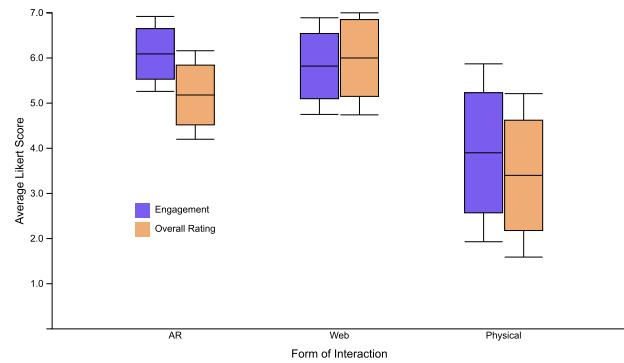


Figure 11: This box plot summarizes the average scores of engagement (purple) and overall rating (orange) based on participants' preferences.

showed too many steps that they could already visualize in their head. The *exemplify* feature was well-liked ($M = 5.55$, $SD = 1.69$), as it helped participants concretize an abstract concept, and some found it easier to understand the notation through interaction (P3, P5). Some participants wanted to see it applied to more complex concepts (P7, P10), while others were more interested in the short-hand formula to solve abstracted notation rather than its detailed expansion (P2). Lastly, participants had mixed feelings about the *dynamic value* feature and did not find it particularly useful compared to the other features ($M = 4.91$, $SD = 1.76$). It was more useful with long solutions (P9) or when combined with other features (P4, P7) but not as much on its own (P6). The results are summarized in Figure 12.

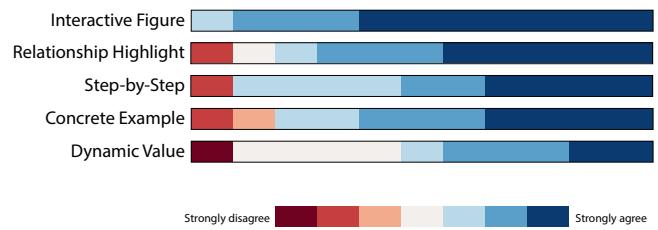


Figure 12: This figure summarize our participants' feedback on the usefulness of the implemented features. From top to bottom, the features are sorted by their average score in descending order.

6.2 Expert Interview

We conducted semi-structured interviews with five math instructors to evaluate the system's potential for educational use. These instructors have teaching experience ranging from 1 to 6 years ($M = 3.4$ years). Three experts teach at the university level (E1, E3, E4), one teaches late elementary to middle school level (E2), and the last teaches from middle school up to early junior university level (E5). During the interviews, we asked them to compare four different educational media: textbooks, videos, interactive websites, and our

AR approach. Each interview lasted approximately 60 minutes, and the experts received compensation of 15 USD.

6.2.1 Method. The goal of the interviews was to gather feedback on the augmented textbook and identify potential areas for improvement. We began by asking the instructors about their experience using textbooks, as well as any challenges or benefits they found in the physical medium. Next, we presented an educational video and an interactive math website covering a topic familiar to the instructors. We asked them to compare these three media as educational resources. Finally, we demonstrated our system through an example authoring walkthrough, allowing the instructors to interact with the system. We gathered their feedback on our AR approach through a series of open-ended questions, focusing on how it compares to existing methods.

6.2.2 Results. Overall, all the experts expressed enthusiasm about our system. They specifically highlighted several key benefits of our approach, including enhanced student engagement, minimal authoring efforts, adaptable content, and potential use for independent learning. Below, we outline our key findings.

Student Engagement through Interactions. All the instructors we interviewed concurred that integrating interactivity into learning materials can effectively encourage students to explore and experiment with concepts at their own pace. They expressed that the interactive nature of our augmented textbook has the potential to transform mundane textbooks into engaging, game-like experiences (E4). Particularly, E5 mentioned that "*the ones that are fascinated by it will play around and try to learn more*". The experts highlighted the value of manipulating content and observing immediate results, especially in calculus, where students grapple with unintuitive concepts like limits. As one instructor noted, "*actually seeing that change happen in real time could be helpful to understand a concept*" (E1). Our augmented reality approach, which combines visual and motor stimuli, may also aid students in retaining information more effectively (E5).

While the experts generally praised the potential benefits of augmented textbooks, most (E1-E4) believed that the available interactions were not always self-explanatory and required guidance. Instructors E3 and E4 specifically emphasized the importance of incorporating built-in guidance to assist them in directing students' attention and interactions. In summary, our system can enhance engagement with traditional materials, enabling seamless integration into existing educational workflows.

Reduced Authoring Effort. During the interviews, instructors revealed that they dedicate a considerable amount of time to crafting materials for their students, often drawing from online resources or textbooks. They also expressed an interest in developing interactive teaching media but frequently lack the necessary skills (E3, E4, E5). Current authoring interfaces can be varied and complex, and instructors may not be well-versed in using them. E1 specifically stated, "*If you ask me to graph a piece wise function on GeoGebra right now, I wouldn't be able to do it*". Our approach holds the potential to substantially reduce the authoring effort needed to create interactive materials. E1 also noted that our approach makes it especially easy to modify existing content rather than starting from scratch. All the experts we interviewed agreed that

they could utilize our tool to produce relevant materials for their students. Overall, our AR approach has the potential to make the creation of interactive materials more accessible and time-efficient for instructors.

On-Demand Adaptable Content. Teachers frequently create customized content to align better with their curriculum and teaching objectives, building upon existing materials. "*I would often just use specific images or drawings from textbooks and papers to explain a concept*" (E4). Our AR approach offers significant support in this area. During the interviews, instructors appreciated the system's ability to select from various augmentation techniques, enabling them to customize the augmented textbook according to their specific needs. This adaptability allows them to tailor the content to their course's learning objectives and their students' requirements. One expert (E1) found that integrating interactive websites into teaching can be challenging due to unfamiliarity with the site's functionalities or difficulties inputting functions into the system. Furthermore, comprehending the website's content and relating it back to the course material can also be challenging (E2). In contrast, our AR approach offers instructors more control over the material they are presenting (E1, E4). This control is particularly beneficial when teaching in front of a class, as customizing the material to suit specific teaching objectives is more desirable than using pre-animated content, such as videos (E1). Overall, our AR approach delivers an efficient and adaptable solution for instructors to create interactive materials that align with their teaching objectives and better engage their students.

Independent Learning. Instructors frequently assign materials for students to review independently, such as videos or online resources (E1, E3). Our AR approach provides an accessible solution for creating customized materials that can be easily shared with students. During the interviews, instructors expressed enthusiasm about the ability to easily share the interactive AR textbook with their students via a web address. Moreover, having access to the materials at any time empowers students to review or advance at their own pace, allowing for more in-depth discussions and knowledge refinement during class time (E5).

7 LIMITATIONS AND FUTURE WORK

Study Limitations

Limitations of Walkthrough Demonstration. While the experts were generally excited about the system, our expert interviews had certain limitations. The majority of the interview was based on a walkthrough demonstration of the AR interface, due to the limited time. Although this provided valuable insights into the system's utility, it did not evaluate the usability of the authoring flow of the system [35]. The experts briefly interacted with our system either through mobile AR (for in-person interviews) and/or desktop interfaces (for remote interviews), but a more comprehensive assessment of the authoring interface is necessary. Additionally, the experts were presented with a pre-defined math textbook and asked how they could incorporate it into their teaching. While this ensured that all experts were exposed to all system features, it would have been interesting to explore how the experts would utilize their own teaching materials.

Real-World Deployment. Although the preliminary user evaluation and expert interviews suggest a potential for using Augmented Math in math education, deploying the system in actual teaching situations would offer further insights into its possibilities. Future work should examine which features instructors would use with their own teaching materials and whether they would discover other features that should be supported. Moreover, evaluating the system in an actual classroom setting with students would be valuable, observing how they interact with the system and exploring its potential benefits for their learning experience.

Improving Usability and Interaction

Supporting More Flexible Authoring Practices. In order to simplify the authoring process and enable non-technical users to create their own explorable explanations, the system relies on OCR and computer vision to automate as much of the process as possible. This includes automatically extracting math formulas and graphs. Although the system accurately detects most content, it occasionally fails. The system also automatically creates *step-by-step hints* based on existing calculations. While most participants in our usability study appreciated these hints, P2 felt that there were too many steps to consume, asking for more flexible control to modify them. To mitigate these soft-errors, future implementations should allow users to manually input equations and graph boundaries. A detection confidence value can also be shown to inform users that they may have to adjust some things. The system also automatically creates *step-by-step hints* based on existing calculations. While most participants in our usability study appreciated these hints, P2 felt that there were too many steps to consume, asking for more flexible control to modify them. Moreover, E2 pointed out that the system should let users not only select from the presented equation but also directly and manually type a new equation to show on a graph. This suggests the need for flexible authoring practice to customize the content for their own needs.

Improving Interaction for Mobile AR. Although participants in the usability study found the AR system to be the most engaging, they rated the usability of the web system higher. Participants encountered difficulties interacting with the AR system due to various reasons, such as small text (P1, P9), shakiness of their hands (P5) and the device itself (P2, P10), and image detection issues (P3). Further work is needed to explore how AR interaction can be improved. For example, future work would explore different possibilities such as magnifying the selected content for better usability or enabling users to take snapshots of the content and interact with it flat on the table instead of directly in AR. Additionally, more work is needed to examine how AR information is presented to the user. For example, the content could be visualized outside the page instead of on top, or AR could be used to simplify the page so that only relevant information is visible.

Beyond Augmenting Math Textbooks

Adapting to Various Topics and Educational Contexts. Our approach has the potential to be applied in subjects beyond mathematics. In language education, for example, *relationship highlight*

and *concrete examples* could be employed to connect abstract grammar rules with examples in the text, such as by highlighting subjects or commas in the text that pertain to the demonstrated rule. In social sciences, *relationship highlight* could also be utilized to emphasize connections between different people and parties or dates and events. In music education, notes and songs on a page could be augmented with sound, enabling users to hear the tone or song. Our approach could also be used to enhance physics education by incorporating physics simulations, allowing users to interact with textbooks in more engaging ways.

Extensibility. Our system relies on computer algebra systems (CAS) like MathJS or SymPy, which are difficult to scale for more complex math. We expect that a different approach would be necessary to support broader topics. For example, an LLM-based approach like ChatGPT + Wolfram Alpha API or Google's Minerva could be an exciting direction to integrate in the future.

Integrating Other AR Interfaces. Finally, we are also interested in exploring other AR interfaces. For example, using projection mapping instead of mobile AR could enhance the teacher's live explanations by augmenting a whiteboard. This approach facilitates different affordances for collaborative learning experiences, enabling students to learn and explore with a group rather than individually. On the other hand, we are also interested in developing explorable explanations for immersive environments through mixed reality headsets. While our presented approach may not be directly applicable due to the lack of precision and resolution of the interaction, it instead opens up new opportunities for immersive explorable explanations beyond 2D interfaces. Future work should continue to investigate this exciting domain by leveraging advanced AR and machine learning techniques for the future of education.

8 CONCLUSION

We introduced Augmented Math, a tool for creating AR-based *explorable explanations* by augmenting static math textbooks. Using OCR and computer vision, our system automatically extracts contents from documents, enabling non-technical users to transform existing math textbooks into personalized, interactive documents without programming. Our proof-of-concept system features five augmentation techniques, based on a taxonomy analysis of existing explorable explanations. We conducted two user studies to evaluate our system, and the results of our studies suggest that our system fosters more engaging learning experiences. Future work includes applying our approach to different domains beyond math textbooks and investigating the potential for classroom use through in-the-wild deployment.

ACKNOWLEDGEMENTS

This research was funded part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant RGPIN-2021-02857. We also thank all of the participants for our user study.

REFERENCES

- [1] Wendy K Adams. 2010. Student engagement and learning with PhET interactive simulations. *Il nuovo cimento C* 33, 3 (2010), 21–32.

- [2] Adobe. [n.d.]. Create augmented reality with AR software - Adobe Aero. <https://www.adobe.com/products/aero.html>. (Accessed on 03/10/2023).
- [3] Nur Ahmad and Syahrul Junaini. 2020. Augmented Reality for Learning Mathematics: A Systematic Literature Review. *International Journal of Emerging Technologies in Learning (iJET)* 15, 16 (August 2020), 106–122. <https://www.learnteclib.org/p/217969>
- [4] Apple. [n.d.]. Reality Composer on the App Store. <https://apps.apple.com/us/app/reality-composer/id1462358802>. (Accessed on 03/10/2023).
- [5] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. 2001. The MagicBook: a transitional AR interface. *Computers & Graphics* 25, 5 (2001), 745–753.
- [6] Nicky Case. [n.d.]. Explorable Explanations. <https://explorables.es/>. (Accessed on 04/03/2023).
- [7] ChanRT. [n.d.]. Visualize It. <https://visualize-it.github.io/>. (Accessed on 04/03/2023).
- [8] Salman Cheema and Joseph LaViola. 2012. PhysicsBook: a sketch-based interface for animating physics diagrams. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, 51–60.
- [9] Zhutian Chen, Wai Tong, Qianwen Wang, Benjamin Bach, and Huamin Qu. 2020. Augmenting static visualizations with paparvis designer. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12.
- [10] Zhutian Chen and Haijun Xia. 2022. Crossdata: Leveraging text-data connections for authoring data documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–15.
- [11] Yu ching Chen. 2019. Effect of Mobile Augmented Reality on Learning Performance, Motivation, and Math Anxiety in a Math Course. *Journal of Educational Computing Research* 57, 7 (2019), 1695–1722. <https://doi.org/10.1177/0735633119854036> arXiv:<https://doi.org/10.1177/0735633119854036>
- [12] Matthew Conlen and Jeffrey Heer. 2018. Idyll: A markup language for authoring and publishing interactive articles on the web. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 977–989.
- [13] Matthew Conlen and Jeffrey Heer. 2022. Fidyll: A Compiler for Cross-Format Data Stories & Explorable Explanations. *arXiv preprint arXiv:2205.09858* (2022).
- [14] Matthew Conlen, Megan Vo, Alan Tan, and Jeffrey Heer. 2021. Idyll studio: A structured editor for authoring interactive & data-driven articles. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 1–12.
- [15] Distill. 2023. Distill. <https://distill.pub/>
- [16] Ruofei Du, Alex Olwal, Mathieu Le Goc, Shengzhi Wu, Danhang Tang, Yinda Zhang, Jun Zhang, David Joseph Tan, Federico Tombari, and David Kim. 2022. Opportunistic Interfaces for Augmented Reality: Transforming Everyday Objects Into Tangible 6DoF Interfaces Using Ad Hoc UI. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 1–4.
- [17] GeoGebra. [n.d.]. GeoGebra - the world's favorite, free math tools used by over 100 million students and teachers. <https://www.geogebra.org/?lang=en>. (Accessed on 03/21/2023).
- [18] Raphael Grasset, Andreas Duenser, Hartmut Seichter, and Mark Billinghurst. 2007. The Mixed Reality Book: A New Multimedia Reading Experience. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems* (San Jose, CA, USA) (*CHI EA '07*). Association for Computing Machinery, New York, NY, USA, 1953–1958. <https://doi.org/10.1145/1240866.1240931>
- [19] Philip J Guo. 2013. Online python tutor: embeddable web-based program visualization for cs education. In *Proceeding of the 44th ACM technical symposium on Computer science education*, 579–584.
- [20] Akash Gupta, Bo Rui Lin, Siyi Ji, Arjav Patel, and Daniel Vogel. 2020. Replicate and reuse: Tangible interaction design for digitally-augmented physical media objects. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12.
- [21] Feng Han, Yifei Cheng, Megan Strachan, and Xiaojuan Ma. 2021. Hybrid Paper-Digital Interfaces: A Systematic Literature Review. In *Designing Interactive Systems Conference 2021* (Virtual Event, USA) (*DIS '21*). Association for Computing Machinery, New York, NY, USA, 1087–1100. <https://doi.org/10.1145/3461778.3462059>
- [22] Andrew Head, Amber Xie, and Marti A. Hearst. 2022. Math Augmentation: How Authors Enhance the Readability of Formulas Using Novel Visual Design Practices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 491, 18 pages. <https://doi.org/10.1145/3491102.3501932>
- [23] Karina Hensberry, Emily Moore, and Katherine Perkins. 2015. Effective student learning of fractions with an interactive simulation. *Journal of Computers in Mathematics and Science Teaching* 34, 3 (2015), 273–298.
- [24] Hiroki Kaimoto, Kyzyll Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1–12.
- [25] Seokbin Kang, Ekta Shokeen, Virginia L. Byrne, Leyla Norooz, Elizabeth Bonsignore, Caro Williams-Pierce, and Jon E. Froehlich. 2020. ARMath: Augmenting Everyday Life with Math Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3313831.3376252>
- [26] Hannes Kaufmann and Dieter Schmalstieg. 2002. Mathematics and Geometry Education with Collaborative Augmented Reality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications* (San Antonio, Texas) (*SIGGRAPH '02*). Association for Computing Machinery, New York, NY, USA, 37–41. <https://doi.org/10.1145/1242073.1242086>
- [27] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 395–405.
- [28] CJ Keller, ND Finkelstein, KK Perkins, and SJ Pollock. 2007. Assessing the effectiveness of a computer simulation in introductory undergraduate environments. In *AIP Conference Proceedings*, Vol. 883. American Institute of Physics, 121–124.
- [29] Mina Khan, Fernando Trujano, Ashris Choudhury, and Pattie Maes. 2018. Mathland: Playful Mathematical Learning in Mixed Reality. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI EA '18*). Association for Computing Machinery, New York, NY, USA, 1–4. <https://doi.org/10.1145/3170427.3186499>
- [30] Tereza Gonçalves Kirner, Fernanda Maria Villa Reis, and Claudio Kirner. 2012. Development of an interactive book with Augmented Reality for teaching and learning geometric shapes. In *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*, 1–6.
- [31] Konstantin Klamka and Raimund Dachselt. 2017. IllumiPaper: Illuminated interactive paper. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 5605–5618.
- [32] Daniel Kunin, Jingru Guo, Tyler Dae Devlin, and Daniel Xiang. [n.d.]. Seeing Theory. <https://seeing-theory.brown.edu/#3rdPage>. (Accessed on 04/03/2023).
- [33] Sam Lau and Philip J Guo. 2020. Data Theater: A live programming environment for prototyping data-driven explorable explanations. In *Workshop on Live Programming (LIVE)*.
- [34] Joseph J. LaViola and Robert C. Zeleznik. 2007. MathPad2: A System for the Creation and Exploration of Mathematical Sketches. In *ACM SIGGRAPH 2007 Courses* (San Diego, California) (*SIGGRAPH '07*). Association for Computing Machinery, New York, NY, USA, 46–es. <https://doi.org/10.1145/1281500.1281557>
- [35] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–17.
- [36] Germán Leiva, Jens Emil Grønbæk, Clemens Nylandsted Klokmose, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2021. Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration. Association for Computing Machinery, New York, NY, USA, 626–637. <https://doi.org/10.1145/3472749.3474774>
- [37] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376160>
- [38] Jingya Li, Erik D. van der Spek, Jun Hu, and Loe Feijis. 2019. Turning Your Book into a Game: Improving Motivation through Tangible Interaction and Diegetic Feedback in an AR Mathematics Game for Children. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (Barcelona, Spain) (*CHI PLAY '19*). Association for Computing Machinery, New York, NY, USA, 73–85. <https://doi.org/10.1145/3311350.3347174>
- [39] Zhen Li, Michelle Annett, Ken Hinckley, Karan Singh, and Daniel Wigdor. 2019. HoloDoc: Enabling Mixed Reality Workspaces That Harness Physical and Digital Content. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (*CHI '19*). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3290605.3300917>
- [40] Mille Skovhus Lundsgaard, Germán Leiva, Jens Emil Sloth Grønbæk, and Marianne Graves Petersen. 2022. ProjectAR: Rapid Prototyping of Projection Mapping with Mobile Augmented Reality. In *Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference*, 1–5.
- [41] Leo A Meyerovich, Arjun Guha, Jacob Baskin, Gregory H Cooper, Michael Greenberg, Aleks Bromfield, and Shriram Krishnamurthi. 2009. Flapjax: a programming language for Ajax applications. In *Proceedings of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications*, 1–20.
- [42] Kyzyl Monteiro, Ritik Vatsal, Neil Chulpongsatorn, Aman Parnami, and Ryo Suzuki. 2023. Teachable Reality: Prototyping Tangible Augmented Reality with Everyday Objects by Leveraging Interactive Machine Teaching. *arXiv preprint arXiv:2302.11046* (2023).
- [43] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. 2018. ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173927>
- [44] University of Colorado Boulder. [n.d.]. PhET: Free online physics, chemistry, biology, earth science and math simulations. <https://phet.colorado.edu/>. (Accessed on 04/03/2023).

- [45] Gallery of Concept Visualization. [n.d.]. Gallery of Concept Visualization. <https://conceptviz.github.io/>. (Accessed on 04/03/2023).
- [46] Katherine Perkins, Wendy Adams, Michael Dubson, Noah Finkelstein, Sam Reid, Carl Wieman, and Ron LeMaster. 2006. PhET: Interactive simulations for teaching and learning physics. *The physics teacher* 44, 1 (2006), 18–23.
- [47] Victor Powell and Lewis Lehe. 2014. Setosa data visualization and visual explanations. <https://setosa.io/>. (Accessed on 04/03/2023).
- [48] Jing Qian, Qi Sun, Curtis Wigington, Han L Han, Tong Sun, Jennifer Healey, James Tompkin, and Jeff Huang. 2022. Dually noted: layout-aware annotations with smartphone augmented reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [49] Shwetha Rajaram and Michael Nebeling. 2022. Paper Trail: An Immersive Authoring System for Augmented Reality Instructional Experiences. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI ’22). Association for Computing Machinery, New York, NY, USA, Article 382, 16 pages. <https://doi.org/10.1145/3491102.3517486>
- [50] Daniel Rodriguez-Clark. 2019. Interactive Maths. <https://www.interactive-maths.com/>. (Accessed on 04/03/2023).
- [51] P. Sarkar, K. Kadam, and J. S. Pillai. 2019. Collaborative Approaches to Problem-Solving on Lines and Angles Using Augmented Reality. In *2019 IEEE Tenth International Conference on Technology for Education (T4E)*. 193–200. <https://doi.org/10.1109/T4E.2019.00-24>
- [52] Jeremy Scott and Randall Davis. 2013. Physink: sketching physical behavior. In *Adjunct Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. 9–10.
- [53] Daniel Smilkov, Shan Carter, D Sculley, Fernanda B Viégas, and Martin Wattenberg. 2017. Direct-manipulation visualization of deep networks. *arXiv preprint arXiv:1708.03788* (2017).
- [54] Hyunyoung Song, Francois Guimbretiere, Tovi Grossman, and George Fitzmaurice. 2010. MouseLight: Bimanual Interactions on Digital Paper Using a Pen and a Spatially-Aware Mobile Projector. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) (CHI ’10). Association for Computing Machinery, New York, NY, USA, 2451–2460. <https://doi.org/10.1145/1753326.1753697>
- [55] James Stewart. 2015. *Single variable calculus: Early transcendentals*. Cengage Learning.
- [56] Hariharan Subramonyam, Steven M Drucker, and Eytan Adar. 2019. Affinity lens: data-assisted affinity diagramming with augmented reality. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.
- [57] Thomas Suselo, Burkhard C. Wünsche, and Andrew Luxton-Reilly. 2021. Using Mobile Augmented Reality for Teaching 3D Transformations. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) (SIGCSE ’21). Association for Computing Machinery, New York, NY, USA, 872–878. <https://doi.org/10.1145/3408877.3432401>
- [58] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. Realitysketch: Embedding responsive graphics and visualizations in AR through dynamic sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 166–181.
- [59] Ryo Suzuki, Gustavo Soares, Andrew Head, Elena Glassman, Ruan Reis, Melina Mongiovi, Loris D’Antoni, and Bjoern Hartmann. 2017. Tracediff: Debugging unexpected code behavior using trace divergences. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 107–115.
- [60] Nikola Ubavić. [n.d.]. GitHub - ubavic/awesome-interactive-math: A curated list of tools that can be used for creating interactive mathematical explorables. <https://github.com/ubavic/awesome-interactive-math>. (Accessed on 04/03/2023).
- [61] Lea Verou, Amy X Zhang, and David R Karger. 2016. Mavo: creating interactive data-driven web applications by authoring HTML. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 483–496.
- [62] Bret Victor. 2011. Explorable explanations.
- [63] Bret Victor. 2011. KillMath.
- [64] Bret Victor. 2012. Learnable programming. *Worrydream. com* (2012).
- [65] Zijie J Wang, Robert Turk, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Polo Chau. 2020. CNN explainer: learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1396–1406.
- [66] Pierre Wellner. 1991. The DigitalDesk calculator: tangible manipulation on a desk top display. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*. 27–33.
- [67] Yuhang Zhao, Yongqiang Qin, Yang Liu, Siqi Liu, Taoshuai Zhang, and Yuanchun Shi. 2014. QOOK: enhancing information revisitation for active reading with a paper book. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*. 125–132.

A ANALYZED WEBSITES

In the following tables (Table 1 and Table 2), we list the interactive websites analyzed in Section 3. For each site, we provide the ID, name, and link, along with the identified design strategy, input method, and domain.

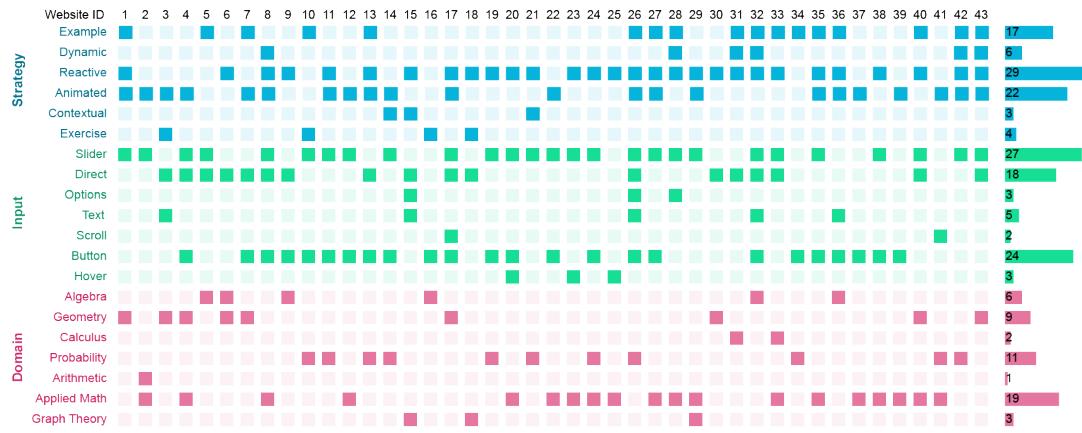


Figure 13: Taxonomy analysis

ID	Name	Link	Strategy	Input	Domain
1	Visualize It - Trigonometric Functions	https://visualize-it.github.io/trig_functions/simulation.html	Animated, Example, Reactive	Slider	Geometry
2	Exponentiation - Explained Visually	https://setosa.io/ev/exponentiation/	Animated	Slider	Applied Math, Arithmetic
3	Relating Area to Circumference	https://www.geogebra.org/m/w5rczf8n#material/bq3gwxxn	Animated, Exercise	Direct, Text	Geometry
4	Seeing circles, sines, and signals	https://jackschaedler.github.io/circles-sines-signals/index.html	Animated	Button, Direct, Slider	Applied Math, Geometry
5	Systems of Linear Algebra	https://textbooks.math.gatech.edu/ila/systems-of-eqns.html	Example	Direct, Slider	Algebra
6	Let's remove Quaternions from every 3D Engine	http://marctenbosch.com/quaternions/	Reactive	Direct	Algebra, Geometry
7	Trigonometry for Games	https://demoman.net/?a=trig-for-games	Animated, Example	Button, Direct	Geometry
8	Back to the future of handwriting recognition	https://jackschaedler.github.io/handwriting-recognition/	Animated, Dynamic Calculation, Reactive	Button, Direct, Slider	Applied Math
9	Slower speed of light	http://gamelab.mit.edu/games/a-slower-speed-of-light/	Reactive	Button, Direct	Algebra
10	The Taxi Cab Problem	http://galgreen.com/TaxiCabProblem/#0	Example, Exercise	Button, Slider	Probability
11	What Happens Next? COVID-19 Futures	https://ncase.me/covid-19/	Animated, Reactive	Button, Slider	Probability
12	Gears – Bartosz Ciechanowski	https://ciechanowスキ/gears/	Animated	Button, Slider	Applied Math
13	Seeing Theory Ch. 1: Basic Probability	https://seeing-theory.brown.edu/basic-probability/index.html	Animated, Example, Reactive	Button, Direct	Probability
14	Going Critical – Melting Asphalt	https://meltingasphalt.com/interactive-going-critical/	Animated, Contextual	Button, Slider	Probability
15	Graphs and Networks	https://mathigon.org/course/graph-theory/introduction	Contextual, Reactive	Direct, Options, Text	Graph Theory
16	Quantum computing for the very curious	https://quantum.country/qcvc	Exercise	Button	Algebra
17	An Interactive Introduction to Fourier Transforms	https://www.jezzamon.com/fourier/index.html	Animated, Reactive	Button, Direct, Scroll, Slider	Geometry
18	The Wisdom and/or Madness of Crowds	https://ncase.me/crowds/	Exercise, Reactive	Direct	Graph Theory

Table 1: Documents analyzed in Study 1 (continued in Table 2). Each website is listed with the ID that identifies it in the paper, its name, link, the design strategy, input method, and domain.

ID	Name	Link	Strategy	Input	Domain
19	Complexity Explorables Horde of the Flies	https://www.complexity-explorables.org/explorables/horde-of-the-flies/	Reactive	Button, Slider	Probability
20	Visualizing the Impact of Feature Attribution Baselines	https://distill.pub/2020/attribution-baselines/	Reactive	Button, Hover, Slider	Applied Math
21	Exploring Bayesian Optimization	https://distill.pub/2020/bayesian-optimization/	Contextual, Reactive	Slider	Probability
22	The Paths Perspective on Value Learning	https://distill.pub/2019/paths-perspective-on-value-learning/	Animated	Button, Slider	Applied Math
23	Computing Receptive Fields of CNN	https://distill.pub/2019/computing-receptive-fields/	Reactive	Hover, Slider	Applied Math
24	A Visual Exploration of Gaussian Processes	https://distill.pub/2019/visual-exploration-gaussian-processes/	Reactive	Button, Slider	Applied Math, Probability
25	Understanding RL Vision	https://distill.pub/2020/understanding-rl-vision/	Reactive	Hover	Applied Math
26	Seeing Theory Probability Distributions	https://seeing-theory.brown.edu/probability-distributions/index.html	Animated, Example, Reactive	Button, Direct, Options, Slider, Text	Probability
27	Firefly Synchronization	https://visualize-it.github.io/firefly_synchronization/simulation.html	Animated, Example, Reactive	Button, Slider	Applied Math
28	Fourier Series	https://visualize-it.github.io/fourier_series/simulation.html	Dynamic Calculation, Example, Reactive	Options, Slider	Applied Math
29	Markov Chains	https://setosa.io/blog/2014/07/26/markov-chains/index.html	Animated, Reactive	Slider	Applied Math, Graph Theory
30	Pythagorean Theorem	https://setosa.io/pythagorean/	Reactive	Direct	Geometry
31	Sine and Cosine	https://setosa.io/ev/sine-and-cosine/	Dynamic Calculation, Example, Reactive	Direct	Calculus
32	Vector Addition	https://phet.colorado.edu/sims/html/vector-addition/latest/vector-addition_en.html	Dynamic Calculation, Example, Reactive	Button, Direct, Slider, Text	Algebra
33	Polynomial Regression	https://visualize-it.github.io/polynomial_regression/simulation.html	Example, Reactive	Direct, Slider	Applied Math, Calculus
34	Random Walks	https://visualize-it.github.io/random_walk/simulation.html	Example	Button	Probability
35	Porous Percolation	https://visualize-it.github.io/porous_percolation/simulation.html	Animated, Example, Reactive	Button, Slider	Applied Math
36	Linear Transformations	https://visualize-it.github.io/linear_transformations/simulation.html	Animated, Example, Reactive	Button, Text	Algebra
37	Mandelbrot Fractal	https://visualize-it.github.io/mandelbrot_fractal/simulation.html	Animated	Button	Applied Math
38	Bernoulli Percolation	https://visualize-it.github.io/bernoulli_percolation/simulation.html	Reactive	Button, Slider	Applied Math
39	Hilbert Curve	https://visualize-it.github.io/hilbert_curve/simulation.html	Animated	Button	Applied Math
40	A Primer on Bézier Curves	https://pomax.github.io/bezierinfo/	Example, Reactive	Direct, Slider	Applied Math, Geometry
41	A visual introduction to machine learning	http://www.r2d3.us/visual-intro-to-machine-learning-part-1/	Animated	Scroll	Applied Math, Probability
42	Conditional probability	https://setosa.io/conditional/	Animated, Dynamic Calculation, Example, Reactive	Slider	Probability
43	Pi (π)	https://setosa.io/ev/pi/	Animated, Dynamic Calculation, Example, Reactive	Direct, Slider	Geometry

Table 2: Documents analyzed in Study 1 (continued from Table 1). Each website is listed with the ID that identifies it in the paper, its name, link, the design strategy, input method, and domain.