

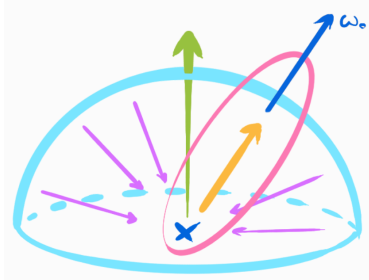
Math Augmentation: How Authors Enhance the Readability of Formulas using Novel Visual Design Practices

Andrew Head
andrewhead@allenai.org
Allen Institute for AI
United States of America

Amber Xie
amberxie@berkeley.edu
UC Berkeley
United States of America

Marti A. Hearst
hearst@berkeley.edu
UC Berkeley
United States of America

$$L_0(\mathbf{x}, \omega_0) = L_e(\mathbf{x}, \omega_0) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_0) L_i(\mathbf{x}, \omega_i) (\omega_i \cdot \mathbf{n}) d\omega_i$$



To find the light towards the viewer from a specific point, we sum the light emitted from such point plus the integral within the unit hemisphere of the light coming from a any given direction multiplied by the chances of such light rays bouncing towards the viewer and also by the irradiance factor over the normal at the point.

Note that incoming light is also computed by that very formula, which makes this exhaustingly recursive.

Figure 1: A formula from computer graphics, visually embellished to improve its readability, from [5] (CC BY-NC-SA 4.0). One author from our interview study created this formula and the accompanying colored diagram and text to teach readers of his blog how to implement the formula in source code. Like many of the formulas analyzed in this paper, this one makes use of color to draw attention to conceptually important expressions in the formula, and to help a reader visually link those expressions to complementary diagrams and prose. Contents of the blog post (formula, prose, and diagram) have been rearranged in this figure to emphasize the formula.

ABSTRACT

With the increasing growth and impact of machine learning and other math-intensive fields, it is more important than ever to broaden access to mathematical notation. Can new visual and interactive displays help a wider readership successfully engage with notation? This paper provides the first detailed qualitative analysis of *math augmentation*—the practice of embellishing notation with novel visual design patterns to improve its readability. We present two qualitative studies of the practice of math augmentation. First is an analysis of 1.1k augmentations to 281 formulas in 47 blogs, textbooks, and other documents containing mathematical expressions. Second is an interview study with 12 authors who had previously designed custom math augmentations (“*maugs*”). This paper contributes a comprehensive inventory of the kinds of maugs that appear in math documents, and a detailed account of how authors’ tools ought to be redesigned to support efficient creation of math augmentations. These studies open a critical new design space for HCI researchers and interface designers.



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9157-3/22/04.
<https://doi.org/10.1145/3491102.3501932>

CCS CONCEPTS

• Human-centered computing → Interactive systems and tools.

KEYWORDS

mathematical notation, authoring, details-on-demand, visual links

ACM Reference Format:

Andrew Head, Amber Xie, and Marti A. Hearst. 2022. Math Augmentation: How Authors Enhance the Readability of Formulas using Novel Visual Design Practices. In *CHI Conference on Human Factors in Computing Systems (CHI '22)*, April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3491102.3501932>

1 INTRODUCTION

With the recent impact and growth of fields like machine learning, people are increasingly trying to educate themselves about the latest advances of math-intensive fields [8]. At the same time, individual authors are rising to the challenge of making mathematical notation more understandable by building creative, sometimes high-profile annotations and augmentations to mathematical notation. In some cases, these authors produce stunning designs despite a lack of appropriate software support tools.

Authors produce these designs to help their readers understand something about a formula that is hard to glean from the notation alone. Take the design in Figure 1, created by an author that was interviewed for this paper. This design embellishes a rendering

equation from computer graphics to help readers understand it. Expressions in the formula are colorized to help readers draw connections between complementary representations of the same idea in notation, prose description, and diagrams. Why might an author show a formula in this way? In the words of this author:

I guess I wrote it for [a younger] version of myself... so when I was in the uni years, if I had had this picture, it would have been so much easier for me to understand the whole concept, right? It took me a long time to understand this formula, or to fully understand it. And if I had had the supporting text with colors and a very simple way to explain it, it would have been nice.

Although authors have been attempting to present notation in novel ways, they currently author these formulas using tools that are either inexpressive or difficult to use. The choices available to authors are math typesetting languages like LaTeX with primitive support for visual design, or graphical editing tools with no built-in support for the structure of notation. Alternatively, authors could use one of an emerging set of mathematical animation tools like *manim* [52], which are capable of creating intricate formula animations, at the cost of a steep learning curve.

This paper envisions a suite of future tools—next-generation markup languages, direct manipulation tools, and automated design assistants—that lower the threshold and widen the walls of authoring tools for math notation. To take a first step towards this vision, this paper answers the questions: *what is it that authors seek to achieve with novel visual designs of notation* and *what are the opportunities for future authoring tools to reduce the friction that authors experience with the current tools*?

To answer these questions, this paper presents an in-depth qualitative analysis of the practice of math augmentation, defined as the embellishment of notation with novel visual designs. First, after a discussion of related work (Section 2), an in-depth content analysis is presented, detailing the sorts of math augmentations (“*maugs*,” pronounced “mogs”) that authors created in 47 documents (Section 3). Second, an interview study is presented with 12 authors of documents containing maugs (Section 4).

The primary conclusion of these studies is that the practice of math augmentation is one that is tedious and yields highly divergent designs. Authors experienced friction working with clunky markup languages, ugly default designs, and tedious graphical editing tools. They desired tools that would help them break free from the limitations of static media, allowing them to show details-on-demand and introduce formulas step-by-step.

A secondary conclusion is that in order to meet authors where they are, tools must support the creation of a diverse complement of maugs. Sixteen kinds of maugs were identified in four different categories: embedded visualizations, text style, annotations (i.e., overlays and underlays), and interactivity. The most pervasive maug was *color*, where color was frequently used to establish visual links between expressions in formulas and nearby prose, notation, diagrams, and code listings. The second most pervasive maug was descriptive labels, which saw considerable diversity in the means by which they were visually associated with expressions.

The primary contribution of this work is a set of eleven design recommendations for future notation authoring tools to support

novel visual presentations, grounded in data from the content analysis and interviews (Section 5). These recommendations lay the groundwork for the design of the next LaTeX, Google Slides, and automated design assistants to enable the pervasive creation of understandable notation.

2 RELATED WORK

This section provides a contemporary account of how formulas might be augmented to influence how they are read. Prior research and design tools are reviewed in the areas of math education, technical writing, visualization, and reading interfaces.

2.1 Math notation

2.1.1 The experience of reading math notation. Reading math notation is difficult. In the math education literature, symbols have been likened to a language with its own grammar and conventions [1]. Outside of formal learning settings, math terminology and notation have been reported as an impediment to learning about advanced technical topics like machine learning [8].

Empirical studies have shown that notation is cognitively demanding to read. While reading math texts, readers frequently shift their attention between formulas and the accompanying prose that describes them [36]. In a study by Österholm [46], readers performed worse on a comprehension task when reading a math text with symbols, rather than a similar text containing solely prose. Notation is less approachable to those who have less experience reading and writing it. In an observational study of both student and expert mathematicians, Shepherd and Sande [57] observed differences in how notation was read. Students might read formulas literally, one symbol at a time, and rely on the text for an interpretation of its meaning. Expert readers, on the other hand, were able to identify idioms in formulas, for instance recognizing a distance computation as the square root of a difference of squares.

When the visual presentation of a formula is altered, readers interpret it differently. Changes to the symbols, spacing, and annotations of a formula have all been shown to influence how a formula is read. For instance, a reader’s expectations of the behavior of an operator, such as its commutativity, is influenced by the operator symbol’s horizontal symmetry [61]. Spacing between operators and their operands can affect how readers interpret the order of operations [39]. The addition of marks such as colors, borders, and arrows can reduce the cognitive load involved in solving algebra problems [63]. Furthermore, the speed with which a formula can be read and later recognized is influenced by whether a formula is well-formed or not, i.e., whether it adheres to the standards of what makes a structurally complete formula such as the presence of two operands for a binary operator [31].

2.1.2 Typesetting and augmenting math notation. Today, myriad tools have been designed to let authors typeset math notation cleanly and then to augment its appearance. Perhaps the most well-known tool is Knuth’s TeX program [35], and its extension, LaTeX. Built into the original TeX is an algorithm for automatically laying out formulas from an input markup language, and the ability to add custom spacing and apply boldface fonts. Subsequently, packages such as “color” [9] and “mathtools” [42] were developed by

the LaTeX community, enabling augmentations to formulas such as the addition of color and labels to formulas, respectively.

As math instruction has increasingly appeared in hypertext and video formats, a new generation of typesetting languages has been developed. KaTeX [16] is an online port of the TeX formula typesetting engine, capable of transforming formula markup into HTML at the time that a page loads. KaTeX implicitly supports a broad array of augmentations by allowing authors to use the full complement of CSS styles on arbitrary expressions. To help authors create interactive formulas, among other types of interactive documents, Victor’s *Tangle* [60] lets authors create reactive formulas, where the values of expressions are computed from input widgets elsewhere on the page. To help authors create math videos, *manim* [52] provides a language for specifying gradual, animated builds of formulas.

Within the HCI community, numerous designs have been proposed to improve how notation is both read and written. Dragunov and Herlocker [15] put forward a vision of such tools, proposing functionality such as the ability to see definitions of expressions on demand, and the ability to step through derivations at multiple levels of detail. Hohman et al. [23] gave a recent account of novel visual and interaction designs for math notation that appeared in their sample of interactive articles. This paper continues in the tradition of these prior papers, providing a comprehensive inventory of the primitives of augmented notation and characterizing how augmented formulas are created today.

Novel affordances for reading math notation have been incorporated into research prototypes. For instance, Head et al. [22] designed ScholarPhi, a reading interface for scientific papers, that allowed readers to look up the definition of symbols in tooltips and in equation diagrams overlaid on top of display equations. Alcock and Wilkinson [2] designed the e-Proofs system for presenting math proofs, wherein authors can fade in and out parts of a proof based on where they want to direct a reader’s attention, and draw arrows to help readers draw connections between related formulas and prose that appear far apart from each other.

The HCI community has also designed systems to help write and manipulate math notation in a variety of forms. Among other features, these systems support the creation of formulas through integrated sketching and formula search [13], performing derivations on formulas using direct manipulation (e.g., pinching two expressions together to add them) [64], and solving algebra problems with complementary symbolic and pictorial representations [32, 53]. These designs suggest a rich space of potential interactions that might help learners deeply engage with math notation.

2.2 Augmented visualizations

Notation is a visual representation of a mathematical idea, one that involves spatial arrangements of textual and symbolic characters. Prior research in visualization provides a vocabulary for describing the different augmentations that can be applied to formulas. For instance, borrowing terminology from Kong et al.’s study of visual cues, or markers that guide a viewer’s attention [37], cues can be “integral” in that they are incorporated into the underlying marks, or they can be “separable” in that the cues are distinct marks. Drawing on the parlance from Hullman et al. [28], annotations can be

“observational” by describing what a viewer can already see, or “additive” by lending additional information for interpreting the visual. The augmentations in this paper are both integral (specifically, embedded visualizations, style, and interactivity augmentations) and separable (annotations). They are mostly additive, helping readers to understand the meaning of math expressions that is not evident from the surface representation of expression alone.

Given the importance of cues and annotations in exposing the meaning of visualizations, the research community has designed tools that aid in their creation with automation (e.g., [7, 28, 58]), domain specific languages (e.g., [17]), and direct manipulation interfaces (e.g., [50, 55]). The purpose of this paper is to review the practice of math augmentation in order to inform the design of tools of each of these types for math notation.

2.3 Augmented technical texts

Math texts are but one type of technical text. Like math texts, technical texts more generally, such as scientific articles, programming tutorials, and data stories, all require readers to undertake the difficult task of integrating information from among multiple representations of complex technical ideas.

Prior research in HCI outlines a rich set of interactions that could be brought to math texts to assist readers. For instance, readers could be aided in quickly navigating between visuals (e.g., tables and charts) and prose that describe the same data [4, 33, 38, 40, 41]. They could be given access to on-demand descriptions of unfamiliar notation [21, 22] and extreme quantities that are hard to visualize [29]. They could be allowed to tinker with parameter choices for quantitative analyses within the document [14]. Given the burden that these interaction techniques place on authors to implement them, authoring tools have been proposed to aid in their implementation (e.g., [11, 40]). This paper provides evidence of which of the above features have analogs in the current practice of math augmentation, and which are desired by authors.

3 STUDY 1. CONTENT ANALYSIS

3.1 Motivation

To understand the expressive potential that maug authoring tools should provide authors, we conducted a content analysis of 47 documents containing maugs. A content analysis methodology is frequently used in studies of visual designs (cf. [34, 50, 54]). The purpose of this analysis was to elicit concrete, actionable, comprehensive guidelines for building tools that create the kinds of maugs that appear in existing documents. The analysis was designed to answer the following research questions:

- (1) What kinds of maugs are used to embellish notation?
- (2) What are the patterns of meaningful variation of each kind of maug?
- (3) Are multiple kinds of maugs used in the same formulas?
- (4) Are maugs applied mostly to simple expressions like identifiers, or are they also applied to more complex expressions?

A description of the content analysis methodology follows. Questions 1 and 2 are answered in Section 3.3.3. Question 3 is answered in Section 3.3.1 and question 4 is answered in Section 3.3.2.

3.2 Methods

3.2.1 Documents. A first sample of documents was formed by selecting documents from the “Awesome mathematical notation design” [25] list that was published as companion material to the article *Communicating with Interactive Articles* [23]. Each document exhibited interesting notation design, or reviewed other documents that did. The collection was iteratively expanded following a snowball sampling approach: each document was reviewed to identify references to other documents that contained augmented notation. This process was repeated until no new documents were identified. Documents that did not contribute novel designs (i.e., those that merely reviewed the practices in other documents) were removed. The sample was then supplemented with 10 research presentations from recent machine learning publications.

The final sample consisted of 47 documents, including 15 web articles, 10 research presentations, 6 chapters from textbooks, 3 Twitter conversations, 3 research papers, 2 Distill articles, 2 videos, 1 lecture presentation, 1 lecture handout, and 4 other documents. Throughout Section 3.3.3, documents exhibiting patterns of math augmentation are referred to by a unique ID, comprised of a letter prefix and a numeric index. For instance, “B3” is the third (“3”) book chapter (“B”) that was analyzed. A listing of documents, their IDs, titles, and URLs appears in Tables 2 and 3 of the appendix.

3.2.2 Units. The unit of analysis in this study was one *augmented formula*, or a formula that was visually altered or marked with the intention of altering how the formula was found, noticed, viewed, read, or understood. Only display formulas, i.e., those that visually stood apart from the prose, figures, and tables, were considered. Hand-sketched formulas were excluded.

The first and second author independently extracted formulas from all documents. The coefficient of reliability was 80.4%, as measured using a metric from Holsti’s text on content analysis [26, p.140]. The two authors reviewed the differences and corrected oversights independently. Remaining differences were resolved through discussion, resulting in 279 formulas. Several formulas were added and removed during the analysis when oversights were noticed, bringing the total number of formulas to 281.

3.2.3 Analysis. A taxonomy of maugs was developed iteratively by the first and second authors on a superset of documents from the sample which contained additional research presentations. An annotation guide was developed with rules for identifying each kind of maug. This guide was revised for clarity as two prospective coders used the guide during training exercises.

Two coders—the first author and another coder—analyzed all formulas according to the guide. Each formula was analyzed for 43 dimensions: 16 dimensions assessing the presence of each kind of maug; 6 counting which types of expressions were augmented (e.g., identifiers vs. numbers vs. others); 9 analyzing label design; 9 analyzing the use of color as a visual link between formulas and other content; 1 analyzing the use of arrows to connect formulas to other content; and 2 dimensions not described in this paper.

Analysis took approximately 50 hours per coder. Each coder made approximately 8k judgments (≈ 281 formulas \times 43 dimensions, minus many dimensions that did not need to be considered as the absence of one dimension meant that another dimension did

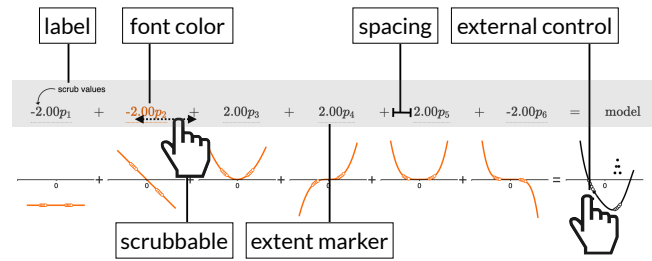


Figure 2: A formula and the kinds of maugs found within it. Shown are two formulas: one that uses symbolic notation (above, highlighted in gray), and a second formula consisting of a sum of function curves (below). The callouts (e.g., “label,” “font color,”) reveal the kinds of maugs that were found in the first formula during analysis. This formula appeared in [18] (CC BY 2.0).

not need to be coded). Krippendorff’s α was used to measure inter-rater reliability after the first pass. The median level of agreement per dimension was .69, with broad variance ($\sigma = .32$, $max = 1$, $min = -0.026$). This revealed systematic disagreements about several dimensions that required resolution. The coders resolved disagreements through extensive discussions. A spreadsheet of the complete analysis results appears in the supplemental material.

Worked example. The following walkthrough demonstrates the analysis of a single formula that appeared in Document D1 (see Tables 2 and 3 for information about the document). The formula was analyzed as containing six kinds of maugs (see Figure 2).

- *Font color*, appearing when an expression is hovered over.
- *Extent markers* (here, underlines) beneath each addend.
- One *label*, reading “scrub values,” above the first addend. The label contains one *phrase*, no *visuals*, and no *notation*.
- *Spacing* around each operator, presumably to align each addend to a function curve in the formula below it.
- *Scrubbing* interactions, where a reader can click and drag on an addend to change the value of its coefficient.
- *External controls*, where a reader can change the values of all coefficients at once by clicking and dragging points in the function curve beneath the “model” expression.

The formula was determined to contain 25 maugs total, belonging to the six kinds described above. To arrive at this total, the following maugs were counted:

- 1 label that augments a numeric literal (“-2.00”).
- 24 maugs across the 6 addends (i.e., the “ $-2.00p_k$ ” expressions), of 4 kinds: extent markers, font colors, scrubbing interactions, and external controls.

Spaces were not included in counts for any formulas due to numerous ambiguities about what could be counted as one unit of irregular spacing. For instance, for the formula above, the space around a “+” sign could be interpreted as one space (i.e., as padding around the “+”), or up to 4 spaces (one on both sides of the “+”, and one on the side of each expression adjacent to the “+”).

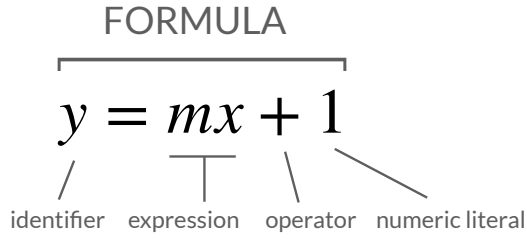


Figure 3: Levels of granularity at which augmentations were counted: identifiers, operators, numeric literals, other expressions apart from the aforementioned types, and entire formulas.

3.3 Results

This section begins with answers to research questions 3 and 4, and concludes with a detailed inventory of the kinds of maugs found in documents (research questions 1 and 2).

3.3.1 Number of Augmentations. When counting maugs, we considered 1 maug to be the application of one *kind* of maug (e.g., font color) to *one expression* (e.g., an identifier like “ x ”). Thus, a formula containing 3 colored identifiers would have 3 maugs.

Across all 281 formulas, 1,182 maugs were counted. Thus, formulas had an average of 4.2 maugs. The median number of maugs per formula was 3 maugs. The number of maugs in any one formula varied greatly ($\sigma = 5.2$, $max = 40$).

Approximately half of formulas featured only one kind of maug. The remaining half (47.7%) were augmented with two or more maugs. The most frequent pairs of kinds of maugs were labels with spacing, and labels with extent markers.

3.3.2 Granularity of Augmented Expressions. Each maug was analyzed as applying to one of five different types of expressions (see a visual reference in Figure 3):

- **Identifiers:** names of mathematical variables or functions, such as “ x ”, “ y_i ”, or “ \sin ”.
- **Operators:** symbols representing mathematical operations, such as “ $+$ ” or the “ Σ ” summation operator.
- **Numeric literals:** numbers, consisting of sequences of digits with optional separators, such as “10,000” or “3.14”.
- **Formulas:** an entire formula.
- **Other expressions:** An expression that is neither the entire formula, nor one of the other types of expressions above.

A minority of maugs applied to identifiers (31.4%). An even smaller minority applied to numeric literals (6.3%), operators (3.3%), and entire formulas (9.1%). Nearly half of augmentations applied to other expressions (49.9%). This category included function calls, operations involving an operator and multiple operands, named constants, and embedded visualizations.

When a formula was augmented, it was frequently the case that some expressions were left unaugmented. For instance, in 68.1% of the formulas where at least one identifier was augmented, at least one other identifier was not augmented at all.

3.3.3 An Inventory of Maugs. Formulas were augmented with 16 kinds of maugs, which are grouped into 4 high-level categories.

		Viz	Style	Annotation	Interactivity												
	Document ID	Geometric object	Visualization	Font color	Font brightness	Space	Extent marker	Border	Background color	Strikethrough	Pointer	Connector arrow	Label	Linked selection	Editable	Scrubable	External controls
Research paper	P1																
	P2																
	P3																
Research slides	S1																
	S2																
	S3																
	S4																
	S5																
	S6																
	S7																
	S8																
	S9																
	S10																
Distill	D1																
	D2																
Web article	W1																
	W2																
	W3																
	W4																
	W5																
	W6																
	W7																
	W8																
	W9																
	W10																
	W11																
	W12																
	W13																
	W14																
	W15																
Video	V1																
	V2																
Book chapter	B1																
	B2																
	B3																
	B4																
	B5																
	B6																
Course notes	N1																
Course slides	L1																
Tweet	T1																
	T2																
	T3																
Other	O1																
	O2																
	O3																
	O4																

Table 1: Summary of results. Each row reveals which kinds of maugs appeared in each document. Documents (rows) are grouped by type, and indexed by a document ID. A mapping from document IDs to titles and URLs can be found in Tables 2 and 3. Kinds of maugs (columns) are grouped by category. Several trends manifest in this table. For instance, it can be seen that font color is used frequently across all types of documents. Interactivity is used mainly in documents hosted as interactive web pages (e.g., web articles, book chapter B3, and Distill articles).

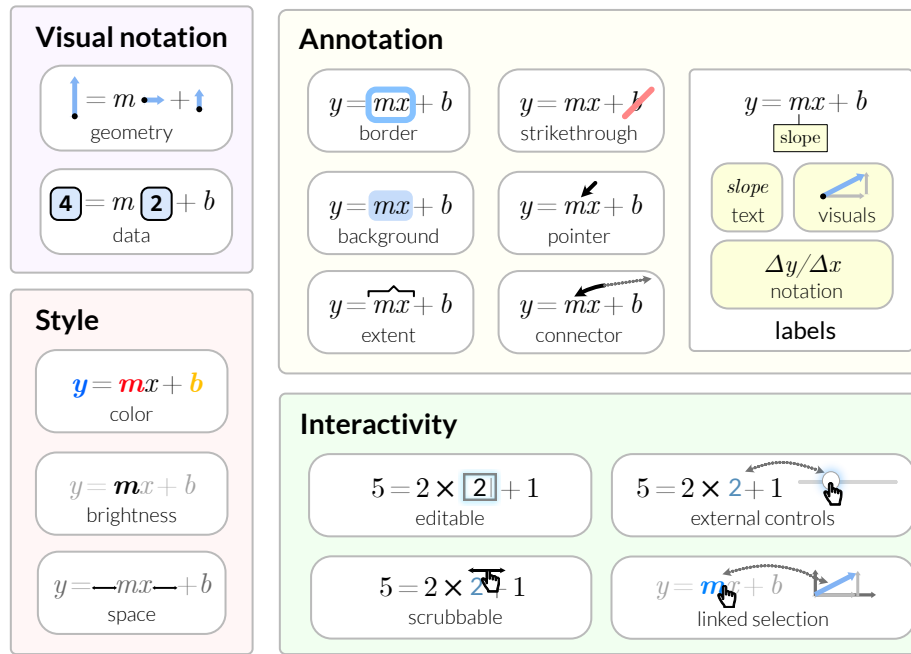


Figure 4: A visual overview of the kinds of maugs observed in the content analysis. Sixteen kinds of maugs were observed, belonging to the categories of visual notation, style, annotation, and interactivity.

These categories roughly correspond to Hullman and Diakopoulos’ “editorial layers” of narrative visualizations (i.e., data, visual representation of data, annotation, and interaction [27]):

- *Visual notation* maugs are those where an expression is entirely replaced with a visualization such as a chart or shape.
- *Style* maugs are those that affect the appearance of text, like the color of symbols and the space between them.
- *Annotation* maugs are marks are overlaid atop or underlaid beneath a formula.
- *Interactivity* maugs are those that allow expressions to be modified interactively via user input.

A visual overview of the kinds of maugs appears in Figure 4. Below, we introduce the kinds of maugs one category at a time. Each kind of maug is described with a name, the percentage of documents in which it was found (e.g., “6.4%”), an example of its use, and patterns of variation in how it was applied to formulas.

Visual notation. Visual notation maugs are visualizations used as expressions. Two kinds of visual notation were observed.

First, formulas could contain *geometric objects*, or shapes such as lines, angles, or circles, as expressions. Geometric objects appeared in 6.4% of documents. They were used in warrants or assertions in visual proofs (B3 (Figure 5A), W6), and other times as mathematical variables where the shape itself conveyed no meaning but rather served as a unique identifier (W14 (Figure 5B)).

Second, formulas could contain *data visualizations* as expressions (8.5% of documents). Data visualizations appeared in formulas representing computations, such as additions, or distance computations,

where the visualizations stood in as example inputs to those computations. Visualizations could be of many kinds, including graphs (D1, S10 (Figure 5C)), images (S8), and pixel colors (W8).

Style. Style maugs affect the format and placement of textual expressions (i.e., Roman symbols, numbers, and combinations thereof).

Font color was the most pervasive kind of maug. It entailed modifying the color of one or more characters in a formula (W14 (Figure 6A)). Font color was altered in nearly half of formulas (41.6%), and in at least one formula in most documents (70.2%).

All formulas were analyzed to determine whether color augmentations (i.e., font color, background color, and colorized extent markers and borders) provided a *visual link*, or a cue suggesting a relationship between an expression and other content. For instance, an expression “ $f(x)$ ” could be colorized green, and the graphical representation of the function $f(x)$ in a nearby graph could also be assigned the color green (see an example in Figure 11).

According to our analysis, most (70.2%) documents used color at least once to establish visual links. Visual links related expressions to both similar (40.4%) and identical (29.8%) expressions in other formulas; to prose (36.2%); to marks (34.0%) and labels (23.4%) in visualizations; and to code in source code listings (6.4%).

A related kind of maug was *font brightness* or *opacity*. This maug could be used to make some expressions more salient by making them easier to see than those surrounding them (D1, Figure 6B). Font brightness was altered in 8.5% of documents.

Space augmentation is the practice of positioning expressions in a formula in a way that would not be done by default by a math

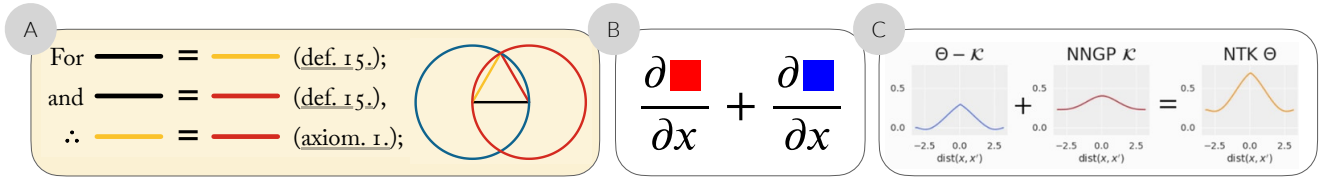


Figure 5: Visual notation. The formulas above exhibit the use of geometric objects in assertions of a visual proof (A, from [51] (CC BY-SA 4.0)); shapes as mathematical variables (B, from [6]); and data visualizations as operands in mathematical computations (C, from [20]).

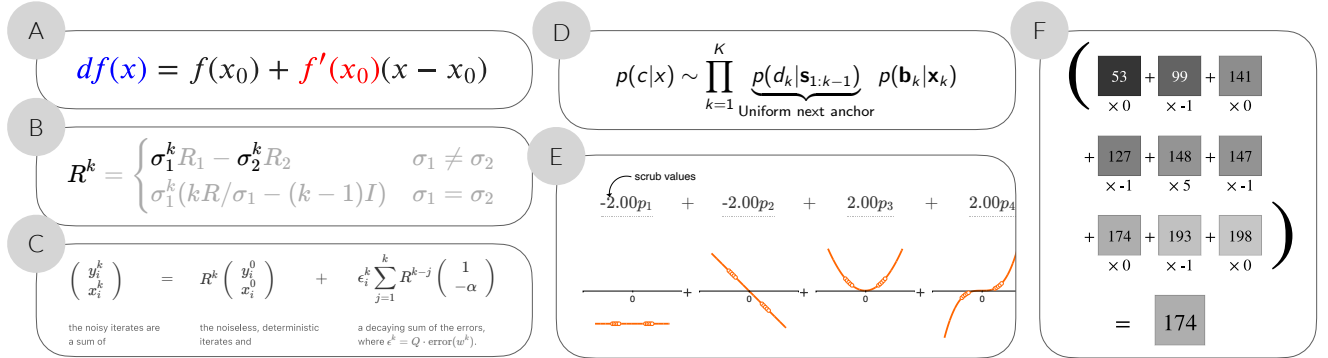


Figure 6: Style augmentations. The formulas above show how expression can be augmented with colored text (A, from [6]), changes to font brightness / opacity (B, from [18]), and irregular spacing (C-F). Space can be used to group symbols into expressions (C, from [18]), align expressions from adjacent formulas (E, from [18]), and arrange expressions in grids (F, from [48]). Sometimes, space between expressions appears to be an unintentional artifact of the authoring tool (D, from [47]).

typesetting engine. Space augmentation was quite common, occurring in 29.8% of documents. Space influenced the appearance of formulas in a few ways. First, it grouped expressions into components representing meaningful concepts, like “decaying sum of the errors” (D1 (Figure 6C), D2). Second, it aligned related expressions in neighboring formulas (D1 (Figure 6E), V1). Third, it was used to convey the gestalt of complex computations. For instance, in a formula from W8 (Figure 6F), products of pixel and filter values were arranged in a 3×3 grid to let a reader perceive the pixels as a cropped segment of the image they came from.

Not all spacing was intentional. Occasionally, formulas contained unexpected spacing on the sides of expressions, often coinciding with the use of labels (e.g., S2 (Figure 6D)). As one of our interviewees confirmed, tools for adding labels sometimes introduce unwanted space. For instance, the extra space in Figure 6D can be explained by the potentially undesirable default behavior of LaTeX’s “\underbrace” macro, which pads expressions with horizontal space to accommodate labels that appear beneath them.

Annotation. Annotation maugs introduce visual marks on top of, or underneath, the text of a formula. Some annotation maugs exhibited little variation, so they are described briefly here before discussing the more complex annotation maugs:

Background color (appearing in 12.8% of documents): the use of color to saturate an otherwise blank background behind an expression. Background colors were used to draw focus to important

expressions in a way that is reminiscent of the use of font colors (B5, W15 (Figure 7A)). Background colors were also used to draw attention to entire formulas (e.g., W13).

Borders (23.4%): containers drawn around expressions. Borders were most frequently drawn around entire formulas (e.g., B4, V2), although on occasion they marked individual expressions (V1 (Figure 7B)). In a few cases, borders were a side effect of a formula appearing in an HTML input box (W6, W12).

Extent markers (29.8%): lines or braces in the margins of a formula along the length of an expression of interest. Extent markers often appeared together with labels (i.e., in 90.0% of formulas in which they appeared), connecting labels to the expressions they described. Extent markers were used in 2 formulas to mark an expression that had been expanded during a derivation (D2 (Figure 7C)).

Strikethroughs (6.4%): marks that cross out an expression to suggest the expression has been eliminated from a formula. Marks used for strikethroughs include slashes (N1, V1 (Figure 7D)) and crosses (V2), and they were colored both red (V1, V2) and black (N1).

Pointer arrows (2.1%): arrows that point to an expression. Pointer arrows were rare, appearing in only one formula (S7 (Figure 7E)).

Two other kinds of annotation maugs stood out as having particularly intricate design spaces. The first was *connector arrows*, arrows that visually linked formulas to other content in a document. Connector arrows appeared in 17.0% of documents. They connected formulas to other formulas they were derived from (W3 (Figure 7G)),

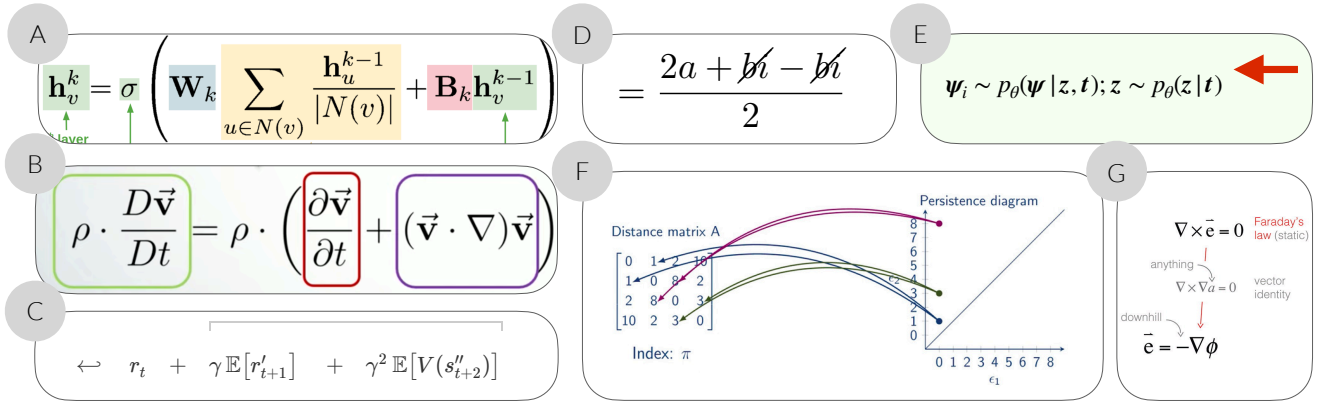


Figure 7: Annotations (excluding label maugs). The formulas above demonstrate augmentation using background colors (A, from [30]), borders (B, from [45]), extent markers (C, from [19] (CC BY 4.0)), strikethroughs (D, from [3]), pointer arrows (E, from [43]), and connector arrows (F, G). Connector arrows need not be unadorned black arrows; the examples here show the use of color (F, from [44]) and labels (G, from [10]) to bring additional meaning to connectors.

text descriptions (V1), and marks in visualizations (S6, (Figure 7F)). Connector arrows could be annotated with labels, and those labels could even contain other formulas (see Figure 7G). Arrows could be colored the same color as the content they connected to (see Figure 7F). A formula could have multiple incoming or outgoing connector arrows, and arrows could connect either to the edges of formulas, or expressions in them (W3).

The second type of annotation maug that had an intricate design space was the *label*. Labels are floating descriptions of expressions. Labels were very common: over 300 labels were counted across all 281 formulas. Labels appeared in 48.9% of documents. Labeled formulas had an average of 3.3 text labels ($\sigma = 2.9$, $\max = 13$), and documents had an average of 6.3 text labels total.

Labels contained myriad types of descriptive content. Most often, they contained text descriptions. In fact, 294 labels contained text. Text was typically short: 75.9% of text labels contained only one phrase, while 6.8% contained complete sentences, and the remainder contained text that was neither a phrase nor a sentence, such as multiple sentences, or notation unaccompanied by textual words. Beyond text, labels contained other types of content: 48 labels in 14 documents contained math notation, and 36 labels in 4 documents included visuals, such as iconography (D2 (Figure 8B)).

Labels were visually associated with expressions in one of three ways: proximity (D1, D2 (Figure 8A)), extent markers (V1 (Figure 8C)), or leader lines. Of the formulas that had labels, 30.0% used extent markers, 45.6% used leader lines, and 8.9% used both.

Although it was rare, sometimes multiple labels applied to a single expression (see “ ρ ” in Figure 8C). Other times, a single label was linked to multiple expressions (W3, W13, W15). Labels could be hierarchical, describing both large expressions and the smaller expressions of which they were composed (Figure 8C). In some cases, labels annotated other labels, rather than expressions, clarifying a description that appeared in another label (B1, T3, W3).

A handful of documents exhibited a pattern we call *pseudo-labels*, where expressions are colored in a way that relates them

to phrases in a nearby floating descriptive sentence (B2, B5 (Figure 8D), O1, W4, W5, W7). This pattern bears a strong resemblance to the use of labels, and its wide usage implies the utility of tools that could help authors create maugs of this kind.

Interactivity. Interactivity maugs alter the appearance and contents of a formula upon user interaction. Because interactivity requires a document to be published in a dynamic medium, only a handful of documents exhibited interactivity (see Table 1). Four kinds of interactivity maugs were observed:

Scrubbable expressions (appearing in 2.1% of documents, i.e., 1 document): Expressions that readers can click and drag to change their value (D1 (Figure 6E)).

Editable text fields (2.1%): expressions that can be edited by changing a value in an input text field (W6).

Linked selections (4.3%): a reader selects or hovers over an expression, which triggers content to be highlighted or selected elsewhere in the document. For instance, in the excerpt from B3 in Figure 5A, when a reader clicks a colored line in a definition or axiom formula, the matching line in the adjacent diagram is highlighted.

External controls (8.5%): expressions that can be controlled using input widgets that appear outside of the formula. External controls influenced expressions in three ways. First, they directly set values of expression. Second, they limited the values that can be assigned to other expressions by other controls. Third, they set hidden variables used in computations of expression values. We encourage the reader to visit document W2, which exhibits all three patterns of control. Input widgets included sliders (D1, W2), drop-down menus (W8), and visualizations with scrubbable points and curves (D1, O4) and content that could be selected by hovering over it (W8).

4 STUDY 2. INTERVIEWS

To understand how tools could help authors create maugs, we conducted an interview study with authors who had previously designed custom maugs.

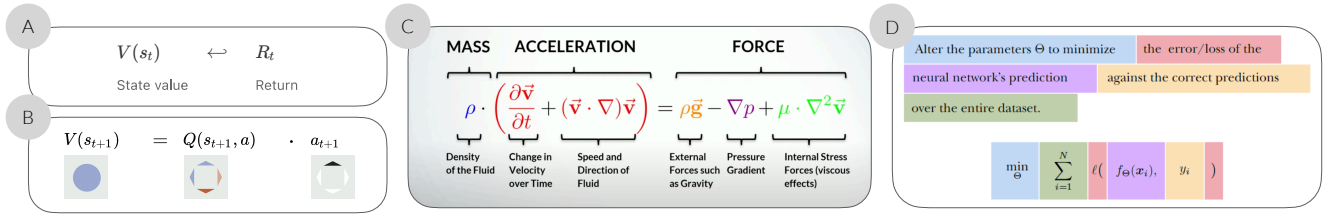


Figure 8: Labels. The formulas above hint at the broad design space of labels. For instance, labels can be associated with expressions using adjacency (A, B, from [19]), extent markers (C, from [45]), or leader lines. Labels can contain text phrases (A, C), visuals (B), or other types of content. Labels can describe overlapping expressions (C). Also shown is an instance of “pseudo-labels,” where a formula is colored in a way that relates expressions to phrases that describe them in a nearby floating sentence (D, from [49]).

4.1 Methods

4.1.1 Participants. From among the documents analyzed in the content analysis (Section 3.2.1), the first author of this paper identified a subset of documents containing particularly elaborate or intriguing augmentations. The authors who created these documents were invited to participate in an interview. Twenty authors were contacted, of whom twelve chose to participate. Authors were highly educated. All held a Bachelor’s degree, and six (half) held Ph.D.s. They were employed as research scientists, software developers, designers, engineering managers, and myriad other occupations. Despite the educational content of many of their designs, only two were formally employed as educators. The maugs they had created had appeared in web articles, book chapters, research papers, and several other document formats.

4.1.2 Protocol. The first author conducted all interviews. First, the interviewer obtained the participant’s informed consent. Then, he asked the participant to describe their occupation and level of education. Then, he opened up a document the participant had authored and directed their attention to one of the augmented formulas. The participant was then engaged in a conversation about how that formula was created, touching upon the following questions (which are paraphrased here for brevity):

- Why did you present the formula in this way?
- How did you create the augmented formula?
- What aspects of the process of creating this augmented formula felt clunky?
- How else would you want to augment the formula if you did not have to do all the work to make it so?

All interviews were held remotely using the Zoom video conferencing software. Interviews lasted 30 minutes. If 30 minutes had passed and the interviewer had questions remaining, he invited the participant to a follow-up session. Five follow-up interviews were held, each between 15 and 45 minutes in duration. Audio recordings were collected for all interviews.

4.1.3 Analysis. Transcripts were made from the audio recordings. They were analyzed to identify participants’ goals, processes, and challenges. The first author developed an initial set of themes by reviewing his personal notes from each interview. Then, he reviewed all transcripts, and further expanded and refined the set of themes. All excerpts that reflected each theme were pulled into a

spreadsheet. The third author helped refine the themes by reviewing all excerpts, determining whether they appeared to represent the themes, and proposing revisions. The final analysis that appears in this paper was agreed upon between the two authors as representative of the themes.

4.1.4 Confidentiality. In the analysis below, authors are referred to by pseudonyms A1–12. Utterances and anecdotes have generally been edited to remove details that could identify either the author or the document they wrote. In some cases where we felt that specific document excerpts and identifiable utterances brought richness to the discussion, we obtained authors’ express permission to include those excerpts and utterances. Document excerpts appear with a citation to the original source.

4.2 Results

This section begins by reviewing authors’ goals in creating maugs and their authoring process, and then discusses the key topic of the section—the challenges authors face when creating maugs. When a theme represents the experience of more than one author, a number appears in parentheses indicating how many interviews support that theme (i.e., “(4)” means “four authors”).

4.2.1 Goals. Authors feared that without scaffolding, readers would be too intimidated to attempt to understand notation. They described notation as something that readers would find “scary,” “dense,” “not that friendly,” “boring,” “big,” “hard to read,” and that notation could “intimidate” and “turn off” readers, or lead them to “freak” or “nope out” (5). Perhaps this was because authors were often writing for audiences who might self-identify as non-mathematical, such as social scientists or qualitative geoscientists (3).

One of the roles that maugs played was to help readers focus on the important parts of big, dense formulas. Authors described this as “breaking up,” “down,” or “out” formulas (3) into “pieces” (3). For instance, A1 described how color could be used to help a reader understand that a formula really includes only a few key conceptual components (see the formula in Figure 9):

But I think color, when mixed with the right notation, can help just give a very quick sense ... you can know here, there’s really just two [things], and zoom in on those two things. So it’s ... it’s like someone has a towel and is wiping your forehead when you’re sweaty ... it’s like, “this is going to be really hard,” they’re like, “no,

local understanding of a model.

GAMs are a generalization of linear models. To illustrate the difference, consider a dataset $D = \{(\mathbf{x}_i, y_i)\}^N$ of N data points, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iM})$ is a feature vector with M features, and y_i is the target, i.e., the response, variable. Let x_j denote the j th variable in feature space. A typical linear regression model can then be expressed mathematically as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_N x_N$$

This model assumes that the relationships between the target variable y_i and features x_j are linear and can be captured in slope terms $\beta_1, \beta_2, \dots, \beta_N$. If we instead assume that the relationship between the target variable and features is smooth, we can write the equation for a GAM [24]:

Figure 9: A passage from [24] where an author used color to help readers see the gestalt of the formula. The author believed that the coloring could help a reader understand the value “ y ” as the combination of just “two things”: the green intercept term “ β_0 ,” and a sum of products between slope terms and feature values made salient with alternating purple and blue color.

no, no, there's only two things you need to know. Stop freaking out. It's not that big a deal...” (A1)

Furthermore, authors believed maugs could empower readers to develop deeper understanding of mathematical ideas by helping them cross-reference the formula with other instructional content:

And so, what I think is deeply underused in the way we communicate at the moment is smart ways of using media to... highlight the aspects of an existing concept that you may have already introduced that you are now talking about and connecting them. So, I see color as kind of like a semantic plane. There are these planes on top of each other, and the concepts pierce the plane, they have all these different aspects that you want to talk about simultaneously in some order. And what color does is, it's a way of slicing through this quite complicated story and then highlighting the different aspects, the relevant almost axes...” (A3).

For many authors, color served as a tool for relating formulas to alternative representations of the same mathematical idea elsewhere in the document. Authors used color to help readers find definitions of expressions in the prose (2), and cross-reference a mathematical expression with illustrative visualizations (4).

4.2.2 Process. The process of augmenting a formula involved three steps and the use of myriad tools.

Step 1. Create formula. Nearly all authors created formulas using the LaTeX language, although nearly all used different environments to write LaTeX. Authors who were creating documents in LaTeX simply wrote formulas as part of their document markup (2). Authors who planned to use drawing tools to create maugs used tools like CodeCogs, MathTeX, and LaTeX2PNG to export rendered LaTeX as vector graphics (2) or high-resolution images (1). Individual authors also reported copying images of formulas from external

sources like Wikipedia, and creating them without LaTeX in drawing programs like Google Slides.

Step 2. Create maugs. There were two dominant approaches for creating maugs: using macros built into LaTeX tools, or creating augmentations using direct manipulation drawing programs. Authors took both approaches, and in fact, some authors did both for different projects. Those who used LaTeX described using it for a limited set of kinds of maugs: color (6), labels (2), and spacing and alignment (1). Eight authors reported using direct manipulation drawing programs such as Google Slides, Microsoft PowerPoint, Adobe Illustrator, Mathcha, and Inkscape. They used these tools to introduce color (3), labels (3), and spacing (2), and two authors described the creation of geometric objects as expressions (2). Both of these approaches came with respective pain points, which are detailed below as authors' challenges.

Step 3. Embed formula in document. Authors integrated formulas into their documents in several ways. The markup for formulas could be embedded directly in the markup code for the document. If authors were writing documents to be hosted on the web in languages like Markdown, they could invoke a LaTeX postprocessor like KaTeX to generate HTML versions of formulas at load time (2). Alternatively, some authors created augmented formulas in their program of choice, exported images of the formulas, and then embedded them in their documents (2).

Two authors implemented interactivity, writing code for custom solutions. One author created a service that let him quickly create calculator widgets with editable text fields for arbitrary formulas. Another author implemented linked selection between visual notation and related marks in nearby diagrams by applying class tags to marks in the notation and diagrams and then programmatically highlighting related content when a user clicked those marks.

4.2.3 Design Inspirations and Sensibilities. When augmenting formulas, authors drew inspiration from external sources. Authors were sometimes versed in principles of visual design; three authors mentioned the work of Edward Tufte (the author of *The Visual Display of Quantitative Data* [59] and other seminal volumes on information visualization) as providing inspiration for how they presented formulas. Authors also drew inspiration from other augmentations they had seen, as well as the types of augmentations that they had seen in source code editor programs, such as syntax highlighting, and the ability to see both CSS selectors and the content that they select side-by-side.

Authors often described their process of design as one that relied on “judgment calls,” “sense,” and careful thought, where it was not clear from the outset how best to embellish a formula (3):

I would just characterize the entire thing as being extremely manual and extremely driven by sense... But I mean, that's partly a function of the fact that I'm try... you know, trying to do something that I haven't really tried to do before, so I'm not totally sure where things should be... (A4).

...making it look like, uh... you know, what I want it to look like, that's the friction... The more complex a diagram is, the harder it is to [rely on computational support]... just because of the number of, you know, small judgment calls you have to make (A12).

First, the order of operations in the numerator don't change with or without the parenthesis, so let's remove them.

$$\frac{\mathbf{x}^T W + \mathbf{b} - \mu}{\sigma} \cdot \gamma + \beta$$

Now let's move γ to the left and apply it to the terms in the numerator. We'll group the two shifts of the bias term \mathbf{b} and the mean μ together.

$$\frac{\mathbf{x}^T W \cdot \gamma + (\mathbf{b} - \mu) \cdot \gamma}{\sigma} + \beta$$

Figure 10: A derivation described in words that perhaps could be better demonstrated through animation. Shown are two steps of a six-step derivation, from [49]. The author of this passage pointed out that the derivation is complex, requiring readers to map transformations described in the text to changes that they can see between successive formulas. For instance, the text explicitly describes how the “ γ ” term (highlighted here for emphasis) can be distributed to each of the terms in the fraction’s numerator. The author believed this derivation could be shown as an animation, where one formula transforms into another, with terms like “ γ ” moving from one position to another.

4.2.4 Challenges. Some augmented formulas could be extremely time-consuming to create. One author reported, for instance, spending an average of one hour per formula; another described spending 20 minutes in LaTeX to get labels to appear just the way he wanted. The challenges in this section are introduced alongside visions that authors proposed of potential improvements to tools that could help address these challenges.

Static media. Authors lamented the limitations of the static media in which they wrote. Few of the authors were writing in media where interactivity or animation were possible. Authors were often constrained to show readers more information about a formula than they wished readers to see. Some authors therefore desired the ability to allow readers to view details on demand (2):

... any good presentation of information should cater to readers and consumers at multiple levels of information so that they can come in and get some of the high-level information quickly... as you use and work with these graphics, they should, like, expose more information to you over time (A6).

Authors also wished for the ability to incrementally introduce complex formulas (5). For instance, A8 described a derivation they had written which required a large amount of prose to describe, and which he felt could be shown more succinctly as a sequence of animated transformations to a single formula (see Figure 10):

... if I look at this, then it's like, “Now let's move ‘gamma’ to the left and apply to the terms in the numerator...” um... [reading more from the document], like that's a “lot” of English to describe what's going on, and it's getting all technical jargon-y, and, like, I don't know how to say it any simpler, but, like, when you do it's

Let's start with a scalar function f , which maps the input x to the output y :

$$y = f(x)$$

Here, f could be a polynomial, exponential, or whatever your favorite kind of mathematical function is.

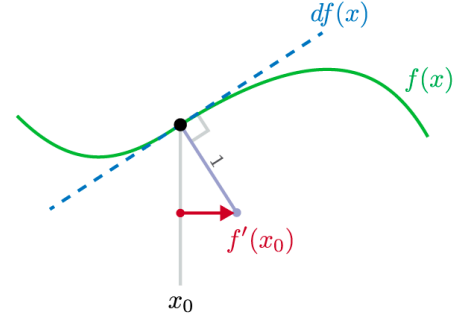


Figure 11: An imagined use of animation to draw connections across analogous representations of a mathematical idea in notation and other forms, from [6]. For instance, in this excerpt from a document A3 wrote, the formula “ $y = f(x)$ ” could be animated in a way that shows a sequence of values plugged in for the variable “ x .” As the value ticks upward, A3 imagined the point on the green curve (representing $(x, f(x))$) moving to reflect the new value of “ x .”

visually, it's just like, “Oh yeah! It sort of goes there” (A8).

While many authors were excited by the idea of more dynamic presentations of formulas, authors’ visions of the ideal presentation varied quite a bit. Authors envisioned tools that could call attention to one set of expressions at a time (2); allow readers to click a formula to query for previous steps in a derivation from which the formula was derived (1); and to visually link formulas and visualizations by animating changes to expressions and marks on visualizations at the same time. This latter idea was described by A3 (see also the accompanying Figure 11):

We can show it algebraically, so we can show “ $f(0)$ ” and then we can, we can animate it so that “ $f(0)$,” that “0” slowly starts ticking up to be “ $f(1)$,” right? And the point in the curve can simultaneously move so that it, it was showing “ $y = f(0)$,” now it's showing “ $y = f(1)$.” The core idea is that you have these different manifestations, they all change simultaneously, and because you can see them both changing, you get a very vivid picture of how wiggling in all the different manifestations corresponds (A3).

Clunky markup languages. While LaTeX was the dominant language for both authoring and augmenting formulas, authors noted that LaTeX was hard to use (2). One challenge was that augmentation required the insertion of macros into the markup of a formula, which made the markup difficult to read (see Figure 12):

Rendered formula:

$$y = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_N(x_N)$$

Markup code:

```
\newcommand{\target}[1]{\textcolor{red}{#1}}
\newcommand{\feature}[1]{\textcolor{blue}{#1}}
\newcommand{\intercept}[1]{\textcolor{green}{#1}}
\newcommand{\shapefunction}[1]{\textcolor{amber}{#1}}
\newcommand{\coeff}[1]{\textcolor{purple}{#1}}
...
$$ \target y
= \intercept{ \beta_0 }
+ \shapefunction{ f_1 }(\feature{x_1}) + \shapefunction{ \coeff
  f_2 }(\feature{x_2}) + \cdots + \shapefunction{ \coeff f_N }(\feature{x_N})
$$
```

Figure 12: A colorized formula and the markup that produces it, from [24]. While the formula is comprised of only 8 identifiers, 5 operators, and 4 colors, it was authored using 414 ASCII characters in LaTeX. The markup indicating the contents of the formula (e.g., “\beta_0”) is tangled with the commands for formatting expressions (e.g., “\intercept”).

That code gets horrible looking, because you’re creating a custom macro for every color. And if you want to have any sort of readable or semantic name, like “purple” or “shape-function-color,” imagine doing that for 10 characters, and you have just, the worst, like it looks so long and complicated to render this small little thing, so that feels unmaintainable (A1).

Furthermore, it could be difficult to keep augmentation-heavy markup syntactically correct, because augmentation might involve managing a large number of nested macros:

I mean, the coding itself was, I think, the struggle, just because matching all the braces up and I’m not using a particularly good IDE for LaTeX, so, you know... just... Yeah, so just getting everything, just getting it to compile is a struggle (A12).

One author tried to use *manim* for presenting augmentation notation. *manim* is a markup language for math animations that provides built-in support for colorizing and animating formulas. While this tool is more expressive than LaTeX, the author reported it had a steep learning curve, estimating it would take a newcomer 1–2 months of use before understanding its conceptual model.

Ugly defaults. When authors used markup languages for authoring maugs of some complexity, they were sometimes dissatisfied with the rendering of those maugs (2). One of the most widely-used packages for diagramming formulas yielded a design that one author described as “horrific” (see Figure 13):

I’ve got these labels here. Uh... these labels are at different elevations on the page. Um... that’s confusing. That’s silly. They should be at the same height. Um, so I don’t like that... [later]... and then the third thing that

immediately stands out to me... is just the bad spacing. Um... like, you have all this extra space that LaTeX is adding around the braces... Like right here, you can barely even tell that “b” is multiplied by “i,” because the “i” is six inches off to the right (A12).

Sometimes, the document compiler (i.e., LaTeX) would unexpectedly alter the layout of a document to accommodate augmentations, which was also undesired:

In previous versions of this, I had tried different types of coloring. For example, using color boxes. Uh... around the text instead... it messes up with like the line heights and stuff (A9).

Tedious graphical editing. When authors used graphical drawing tools, they experienced all of the problems intrinsic to software for graphical editing in general. For those experimenting with novel presentations, this was described as a particularly tedious part of the process (4). Tasks such as aligning content, adding augmentations, managing copies of content, and moving groups of related content, were described as tedious to perform.

Just the manual-ness of this process... And it’s not just the inconvenience of that, but it’s obviously highly error-prone. Like it would be super easy for me to leave the incorrect label somewhere, or, just have something that’s totally wrong on there, because I overlooked the fact that I’d actually updated something on there, or copy-and-pasted the label from another page, so I got the same formatting... Uh, and then just forget to edit it after I’ve pasted it (A4).

One approach to mitigating this tedium was simply to be less precise. One author described how the tools he used, in this case PowerPoint instead of Illustrator, lessened the temptation to achieve a pixel-perfect layout. Another author speculated that some authoring effort could be alleviated if tools were aware of the semantics of formulas and their augmentations:

In InkScape... nothing’s attached to anything, right? So everything’s floating. There’s no semantic meaning to anything..... What it would it look like in Google Slides if you could attach pointing labels to things, and now when I move the thing, the label moves? (A4)

Hitting the sweet spot of augmentation. Authors were faced with complex design decisions around augmentation, namely how to augment formulas in a document while keeping the document readable as a whole. This led to a number of open questions about effective augmentation design. For instance, one author was concerned with using labels to describe expressions, thinking that they imposed on readers a choice of which labels to read and in what order. For this reason, this author preferred to use pseudo-labels, relating expressions to prose using color, because it meant that readers would not have to choose in what order to read the descriptive text:

... where [among the labels] do you start reading? Because it’s no longer really like a left-to-right thing, like a sentence is. And if you’re reading a sentence, you know exactly where to start reading, because that’s how

$$\begin{array}{c}
 \text{rectangular/Cartesian forms} \qquad \qquad \qquad \text{polar/mod-arg forms} \\
 (a, b) = \underbrace{a}_{\text{real component}} + \underbrace{b \cdot i}_{\text{imaginary component}} = r \angle \theta = r i^{\frac{\theta}{\pi/2}} = \underbrace{r}_{\text{radius}} \cdot \left(\underbrace{\cos \theta}_{\text{real component}} + i \cdot \underbrace{\sin \theta}_{\text{imaginary component}} \right) = \underbrace{r}_{\text{radius}} \underbrace{e^{i \cdot \theta}}_{\text{angle}} = \underbrace{r}_{\text{radius}} \underbrace{e^{i \cdot \theta}}_{\text{exponential/Euler form}}
 \end{array}$$

Figure 13: An example of a formula for which the default layout algorithms failed to produce a satisfactory layout for an author, from [3]. A12 named issues with this presentation including unwanted horizontal spacing (for instance, a large distance between the second “b” and the subsequent “·” operation), and labels that do not vertically align with each other.

sentences work. They go from left-to-right, top-down, unless it's another language (A8).

Authors were also sensitive to the possibility that there could be too much color added to a document (3):

... it was tricky because we tried to minimize the amount of coloring, because of that, we used color in other diagrams and in other ways. So one, I used to actually use more color, but I found that, like, it was... Some of the information wasn't as important as others, so I got rid of a lot of it, yeah (A9).

Sensitivity to over-coloring was representative of a larger concern that one author described as managing the “budget” of augmentations, or recognizing that one can only add so many augmentations before they begin to conflict with each other:

But there's an allocation problem... you've got a little budget of variables, planes, that you have access to, but you can't use them for more than one thing. So once I've chosen what to use color for, I can't really use it again... So you've got to juggle how you spend your budget of stuff (A3).

Making cross-cutting style changes. Authors would use similar maugs for similar formulas in the same document, like using a single color for related expressions in multiple formulas (5). When a document was very long, managing styles that cut across the document could be prohibitively difficult. One author of a very long document had colorized formulas using a graphical editing tool. While he made an effort to use similar colors for related expressions that appeared in a subset of formulas, he thought interface support was necessary to consistently colorize related expressions the same way across the whole document:

... another level of effort too far and would really need like software support, is to make the colors correspond to the nature of the equation. So... like if, maybe I've got the gradient of “theta” there, maybe that's always gray in every equation. Or everything involving some kind of gradient calculation stays that color (A8).

If an author was augmenting formulas with LaTeX, they could leverage macros to experiment with cross-cutting style changes, encoding design decisions in macros and testing out alternatives by changing their expansions. For instance, one author creating macros for the colors assigned to classes of expressions in a document where formulas were rendered using the KaTeX library:

So there's macros for colors. Thank God. I use a lot of macros. I define my own macros to make it easier to even write this stuff, just because certain conventions happen again and again... So what I've gotta do, is I've got to make sure the way I've authored the document, that's a switch that I can flip really easily at any point (A3).

Additional Feature Requests. When given license to consider future tools for presenting math notation, authors suggested additional capabilities they wished for their tools to have. These include new types of maugs (e.g., commands for programmatically changing the sizes of expressions, novel visual representations of indices for matrices and vectors), the ability to more easily reuse augmented formulas outside of their original document, and interactive simulations and computations.

5 DESIGN RECOMMENDATIONS

In this section, we derive a set of recommendations for what features tools should provide to help authors create augmented notation, connecting the recommendations to evidence from the two studies. We note that the recommendations are not requirements, in that a useful authoring tool need not satisfy all of these recommendations. Rather, each one indicates an opportunity for a future tool to lower the threshold for authors who would want to create maugs but under current circumstances would not do so.

The recommendations support three augmentation approaches: next-generation markup languages that succeed LaTeX, graphical editors that let authors augment formulas with direct manipulation; and intelligent design assistants that automatically augment formulas on an author's behalf. Below, recommendations are first presented that apply to all three of these augmentation approaches, followed by more specific recommendations for each approach.

5.1 General recommendations

R1. Support multiple kinds of maugs in each formula. A majority of documents (68.0%) and nearly half of all formulas (47.7%) used more than one kind of maug. Within a formula, different kinds of maugs could be used to reinforce each other, such as introducing space in a formula in a way that allowed labels to be visually grouped with expressions without leader lines. To support these authoring patterns, tools should make it possible to apply multiple kinds of maugs in a single formula.

R2. Help authors color formulas and the content they relate to. Color was used to visually link formulas to other content in most (70.2%) documents. Color related expressions to descriptions in prose, expressions in other formulas, code, and the contents of visualizations. Our informants used color to help readers access analogous representations of the same mathematical idea in notation, prose, and visuals. Tools may be able to assist authors by helping them pick and apply color palettes across the many formulas and related content in a document.

R3. Support details-on-demand and animations. Our informants felt limited by the constraints imposed by the static media in which they authored, and believed that the presentation of notation could be improved with interactivity, animation, and details-on-demand. For instance, authors desired the ability to show detailed descriptions of expressions on-demand rather than all at once, and believed that derivations might be more effectively demonstrated with animations on a single formula than with a conventional prose description of the derivation.

R4. Provide input widgets for setting, constraining, and indirectly influencing expressions values. Interactivity was uncommon in augmented formulas, perhaps due to the static nature of many documents, or the challenge of implementing interactivity. That said, the most common form of interactivity was the use of external controls to influence expression values, appearing in 8.5% of documents. Input widgets served three purposes: directly setting values in formulas, setting limits on the values they could take on, and setting hidden variables used in computations. Toolkits that support interactive computation of expression values may wish to let readers control expressions in each of these three ways.

R5. Respect the document's style. Maugs are not shown in isolation; rather, they are a part of documents that have may have been painstakingly formatted. One informant described a problematic augmentation macro that altered the spacing of lines. Augmentation tools should integrate nicely into the editing workflow for an entire document, and one important part of a clean integration of an augmentation tool is to respect the document's existing style.

5.2 Markup language design

R6. Make markup languages easy to read and write. Our informants found it difficult to read and edit markup code for augmented formulas due to the clutter that was introduced by macros for coloring and labeling expressions. Markup languages could be redesigned to separate code for augmenting a formula from the specification of the underlying formula. Such an approach has seen widespread use in language design in general, from the widely used HTML and CSS, to HCI systems such as Penrose [62], which offer complementary languages for specifying content and style of visual material.

R7. Let authors embed visualizations in formulas. In 8.5% of documents, augmented formulas contained visualizations as expressions such as geometric objects, charts, and images. Markup languages should let authors embed visualizations like these in formulas.

R8. Support a rich design space for labeling expressions. In the reviewed documents, many labels contained math notation (29.8%), and some contained visuals like icons (8.5%). Labels sometimes

applied to nested expressions, describing a formula at multiple levels of granularity. In some cases, one expression had multiple labels, or one label applied to multiple expressions. Markup languages for augmented notation should give authors flexibility in how they assign labels to expressions, and what content goes into a label.

5.3 Graphical editor design

R9. Bridge the divide between LaTeX and vector graphics. Informants frequently created formulas using LaTeX, and then exported them as vector graphics or bitmaps so that they could be edited in graphical editing programs. Some editors (e.g., PowerPoint) let authors create bitmaps from LaTeX within the editor, though none of our informants mentioned using a graphical editor that let them create vector graphics from LaTeX. A related issue is the tedium of graphical editing. Perhaps future tools could make easier for authors to align, augment, and move expressions if the tools were aware of the underlying structure of formulas.

5.4 Automated tool design

Could automated tools augment formulas with minimal author input, for instance by automatically coloring expressions and the corresponding text that describes them? Perhaps, though such tools would first have to address the following challenges:

R10. Identify and augment meaningful expressions. An automated tool may need to identify expressions that are worth augmenting. Our content analysis makes it clear that it is not enough to augment mathematical identifiers like “ x ” or “ t_k ”. Only a minority of maugs applied to such identifiers (31.4%), and even fewer applied to numeric literals, operators, and entire formulas. Rather, nearly half of augmented formulas were expressions that could not be classified as any of the above (49.9%), including function calls, and operations involving multiple operands. To match patterns of human augmentation, automated tools would need to identify meaningful, significant higher-level expressions that merit being augmented.

R11. Limit the number and variety of augmentations. Our informants believed that the effectiveness of augmentations tapered off when notation was augmented without care. For instance, authors held beliefs that a document could be colorized too much, that color could not be reused for more than one concept, or even that labels could induce a cognitive load greater than colorizing a formula the same colors as prose that described it. While further studies are needed to characterize the complex tradeoffs between these design decisions, it is clear that a good automated tool would provide the right amount of the right kinds of augmentation, rather than attempting to augment as many expressions as it can.

6 LIMITATIONS AND FUTURE WORK

6.1 Limitations

The generalizability of the findings from the content analysis is limited by the selection of documents. Several categories of documents (e.g., videos, course notes) saw only light representation. Therefore, we do not claim that theoretical saturation (cf. [12, page 263]) was achieved. The under-representation of some categories of documents suggests that the inventory of maugs may not be complete. For instance, animation is not in the inventory, even though

it was of considerable interest to authors, because it appeared in only one document. It should also be noted that the frequencies of maugs may not be representative of all math documents.

While authors offered their beliefs about how augmentations could help readers, our interview study does not provide clear guidance how authors should *ideally* augment formulas for their readers. Follow-up studies would be needed to assess how various augmentations affect the reading experience, including both the benefits and drawbacks of augmentation.

6.2 Future Work

6.2.1 Authoring tools for maugs. The recommendations in Section 5 and the inventory of maugs from Section 3 provide a partial specification of capabilities that maug authoring tools should have, whether that tool is a markup language, a graphical tool, an automated assistant, or something else. Visualization grammars such as Vega-Lite [56] and Canis [17] have enabled the creation of new tools for visualization generation and authoring. Could similar tools be developed for augmenting mathematical notation? Could TeX be extended with primitives for embedded visualizations, style, annotations, and interactivity? Perhaps entirely new languages are needed to support the conflicting goals of augmenting formulas and producing markup that is readable, as well as for authoring designs that animate formulas or show details on demand.

6.2.2 Characterizing the usability of augmented math documents. Even if authors have tools for augmenting formulas, they need an understanding of effective design patterns if they are to improve the reading experience. Further research is needed to explore which kinds of maugs are the most effective. Questions to be investigated include: Should expressions be defined by colorizing or labeling? Given a formula that contains dozens of symbols, what is the maximum number of symbols that could be labeled before the labels overwhelm a reader? Answering these questions may require empirical studies assessing how outcomes like learning, cognitive load, and memorability are affected by different design patterns.

7 CONCLUSION

Two qualitative studies were presented to shed light on how authors embellish the presentation of mathematical notation to improve its readability. The first study, a content analysis, revealed that authors created highly divergent presentations of formulas, incorporating embedded visualizations, text styling, space, annotations, and interactivity. The majority of documents made use of color to visually link expressions to other content such as prose descriptions, related formulas, diagrams, and source code. The second study, an interview study, revealed that the authoring process is tedious. Authors experienced significant challenges using current tools. This work provides an inventory of features that future authoring tools should provide to make visual and interactive embellishments both expressive and usable, summarized in eleven recommendations for the design of future markup-based, graphical editing, and automated authoring tools.

ACKNOWLEDGMENTS

Dongyeop Kang contributed to the design of Study 1, and Grace Chang served as the second coder in the analysis. We thank the

anonymous interviewees of Study 2 for their generous contributions of time and expertise, and the anonymous reviewers for their thoughtful comments. This research received funding from the Alfred P. Sloan Foundation and the Allen Institute for AI.

REFERENCES

- [1] Thomasenia Lott Adams. 2003. Reading mathematics: More than words can say. *The Reading Teacher* 56, 8 (2003).
- [2] Lara Alcock and Nicola Wilkinson. 2011. e-Proofs: Design of a Resource to Support Proof Comprehension in Mathematics. *Educational Designer* 1, 4 (2011).
- [3] Andrew M.H. Alexander. *Exponential Improvements*. <http://www.andrusia.com/math/complex-numbers/eulers-identity.pdf> Last accessed December 27, 2021.
- [4] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. 2019. Elastic Documents: Coupling Text and Tables through Contextual Visualizations for Enhanced Document Reading. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019).
- [5] Sergio Basurco. *Isaac Hayes Wallpaper Generator - Volumetric light scattering, 1 of 2*. <https://chuckleplant.github.io/2017/05/28/light-shafts.html> Last accessed December 27, 2021.
- [6] Tali Benyon. *Build your own Tensorflow*. <https://taliesin.ai/projects/edu/indaba-2019/> Last accessed December 27, 2021.
- [7] Chris Bryan, Kwan-Liu Ma, and Jonathan Woodring. 2017. Temporal Summary Images: An Approach to Narrative Visualization via Interactive Annotation Generation and Placement. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017).
- [8] Carrie J. Cai and Philip J. Guo. 2019. Software Developers Learning Machine Learning: Motivations, Hurdles, and Desires. In *Proceedings of the Symposium on Visual Languages and Human-Centric Computing*.
- [9] David P. Carlisle. *The color package*, CTAN-Archive.
- [10] Rowan Cockett, Lindsey Heagy, and Doug Oldenburg. 2016. Pixels and their neighbors: Finite volume. *The Leading Edge* 35, 8 (2016).
- [11] Matthew Conlen and Jeffrey Heer. 2018. Idyll: A Markup Language for Authoring and Publishing Interactive Articles on the Web. In *Proceedings of the Symposium on User Interface Software and Technology*.
- [12] Juliet Corbin and Anselm Strauss. 2008. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3rd ed.). SAGE Publications.
- [13] Yancarlos Diaz, Gavin Nishizawa, Behrooz Mansouri, Kenny Davila, and Richard Zanibbi. 2021. The MathDeck Formula Editor: Interactive Formula Entry Combining LaTeX, Structure Editing, and Search. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [14] Pierre Dragicevic, Yvonne Jansen, Abhraneel Sarma, Matthew Kay, and Fanny Chevalier. 2019. Increasing the Transparency of Research Papers with Explorable Multiverse Analyses. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [15] Anton N. Dragunov and Jonathan L. Herlocker. 2003. Designing intelligent and dynamic interfaces for communicating mathematics. In *Proceedings of the International Conference on Intelligent User Interfaces*.
- [16] Emily Eisenberg and Sophie Alpert. *KaTeX*. <https://katex.org> Last accessed September 16, 2020.
- [17] Tong Ge, Yue Zhao, Bongshin Lee, Donghao Ren, Baoquan Chen, and Yunhai Wang. 2020. Canis: A High-Level Language for Data-Driven Chart Animations. *Computer Graphics Forum* 39 (2020).
- [18] Gabriel Goh. 2017. Why momentum really works. *Distill* 2, 4 (2017). <http://distill.pub/2017/momentum>
- [19] Sam Greydanus and Chris Olah. 2019. The Paths Perspective on Value Learning. In *Distill*. <https://distill.pub/2019/paths-perspective-on-value-learning>
- [20] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. *Bayesian Deep Ensembles via the Neural Tangent Kernel*, *NeurIPS Poster Presentation*. https://nips.cc/virtual/2020/protected/poster_0b1ec366924b26fc98fa7b71a9c249cf.html Last accessed December 30, 2021.
- [21] Andrew Head, Codanda Appachu, Marti A. Hearst, and Björn Hartmann. 2015. Tutorons: Generating context-relevant, on-demand explanations and demonstrations of online code. In *Proceedings of the Symposium on Visual Languages and Human-Centric Computing*.
- [22] Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. 2021. Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [23] Fred Hohman, Matthew Conlen, Jeffrey Heer, and Duen Horng Chau. 2020. Communicating with Interactive Articles. *distill.pub*.
- [24] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. 2019. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.

- [25] Fred Hohman and other contributors. 2020. Awesome Mathematical Notation Design. <https://github.com/fredhohman/awesome-mathematical-notation-design>
- [26] Ole Holsti. 1969. *Content Analysis for the Social Sciences and Humanities*. Addison-Wesley.
- [27] Jessica Hullman and Nicholas Diakopoulos. 2011. Visualization Rhetoric: Framing Effects in Narrative Visualization. *IEEE Transactions on Visualization and Computer Graphics* 17 (2011).
- [28] Jessical Hullman, Nicholas Diakopoulos, and Eytan Adar. 2013. Contextifier: automatic generation of annotated stock visualizations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [29] Jessica Hullman, Yea-Seul Kim, Frances Nguyen, Lauren Speers, and Maneesh Agrawala. 2018. Improving Comprehension of Measurements Using Concrete Re-expression Strategies. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [30] Ankit Jain, Isaac Liu, Ankur Sarda, and Piero Molino. *Food Discovery with Uber Eats: Using Graph Learning to Power Recommendations*. <https://eng.uber.com/uber-eats-graph-learning/> Last accessed December 30, 2021.
- [31] Anthony R. Jansen, Kim Marriott, and Greg Yelland. 1999. Perceiving structure in mathematical expressions. In *Proceedings of the Conference of the Cognitive Science Society*.
- [32] Seokbin Kang, Ekta Shokeen, Virginia L. Byrne, Leyla Norooz, Elizabeth M. Bonsignore, Caro Williams-Pierce, and Jon Froehlich. 2020. ARMath: Augmenting Everyday Life with Math Learning. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [33] Dae Hyun Kim, Enamul Hoque, Juho Kim, and Maneesh Agrawala. 2018. Facilitating Document Reading by Linking Text and Tables. In *Proceedings of the Symposium on User Interface Software and Technology*.
- [34] Hwiyeon Kim, Juyoung Oh, Yunha Han, Sungahn Ko, Matthew Brehmer, and Bum Chul Kwon. 2019. Thumbnails for Data Stories: A Survey of Current Practices. In *Proceedings of the IEEE Visualization Conference*.
- [35] Donald Ervin Knuth. 1986. *Computers & Typesetting, Vol. A: The TeXbook*. Addison-Wesley Longman Publishing Co., Inc.
- [36] Andrea Kohlhasse, Michael Kohlhasse, and Taweechai Ouyornkochagorn. 2018. Discourse Phenomena in Mathematical Documents. In *Intelligent Computer Mathematics*, Florian Rabe, William M. Farmer, Grant O. Passmore, and Abdou Youssef (Eds.).
- [37] Ha-kyung Kong, Wenjie Zhu, Zhicheng Liu, and Karrie Karahalios. 2019. Understanding Visual Cues in Visualizations Accompanied by Audio Narrations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [38] Nicholas Kong, Marti A. Hearst, and Maneesh Agrawala. 2014. Extracting references between text and charts via crowdsourcing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [39] David Landy and Robert L. Goldstone. 2007. Formal notations are diagrams: Evidence from a production task. *Memory & Cognition* 35 (2007).
- [40] Shahid Latif, Zheng Zhou, Yoon Kim, Fabian Beck, and Nam Wook Kim. 2021. Kori: Interactive Synthesis of Text and Charts in Data Documents. In *Proceedings of the IEEE Visualization Conference*.
- [41] Junhua Lu, Wei Chen, Hui Ye, Jie Wang, Honghui Mei, Yuhui Gu, Yingcai Wu, Xiaolong Luke Zhang, and Kwan-Liu Ma. 2021. Automatic Generation of Unit Visualization-based Scrollytelling for Impromptu Data Facts Delivery. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*.
- [42] Lars Madsen, Will Robertson, and Joseph Wright. *The mathtools package*, CTAN-Archive, ctan.org.
- [43] Chenlin Meng, Lantao Yu, Yang Song, Jiaming Song, and Stefano Ermon. *Autoregressive Score Matching*, NeurIPS Poster Presentation. https://nips.cc/virtual/2020/protected/poster_4a4526b1ec301744aba9526d78fcb2a6.html Last accessed December 30, 2021.
- [44] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. *Topological Autoencoders*, ICML Poster Presentation. <https://icml.cc/virtual/2020/poster/5851> Last accessed December 30, 2021.
- [45] Jousef Murad. *Derivation of the Navier-Stokes Equations*. <https://www.youtube.com/watch?v=zWdnf3Uh1RE> Last accessed December 30, 2021.
- [46] Magnus Österholm. 2006. Characterizing Reading Comprehension of Mathematical Texts. *Educational Studies in Mathematics* 63 (2006).
- [47] Ari Pakman, Yueqi Wang, Catalin Mitelut, and JinHyung Lee. *Neural Clustering Processes*, ICML Poster Presentation (slide 27). <https://icml.cc/virtual/2020/poster/6425> Last accessed December 30, 2021.
- [48] Victor Powell. *Image Kernels Explained Visually*. <https://setosa.io/ev/image-kernels/> Last accessed December 30, 2021.
- [49] Edward Raff. 2020. *Inside Deep Learning: Math, Algorithms, Models*. Manning.
- [50] Donghao Ren, M. Brehmer, Bongshin Lee, Tobias Höllerer, and E. Choe. 2017. ChartAccent: Annotation for data-driven storytelling. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*.
- [51] Nicholas Rougeux. *Byrne's Euclid*. <https://www.c82.net/euclid/book1> Last accessed December 30, 2021.
- [52] Grant Sanderson. *Manim*. <https://3b1b.github.io/manim/>
- [53] Nazmus Saquib, Rubaiat Habib Kazi, Li-yi Wei, Gloria Mark, and Deb Roy. 2021. Constructing Embodied Algebra by Sketching. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*.
- [54] Alper Sarikaya, M. Correll, L. Bartram, Melanie K. Tory, and Danyel Fisher. 2019. What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization and Computer Graphics* 25 (2019).
- [55] Arvind Satyanarayan and Jeffrey Heer. 2014. Authoring Narrative Visualizations with Ellipsis. *Computer Graphics Forum* 33 (2014).
- [56] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 23.
- [57] Mary D. Shepherd and Carla V. D. Sande. 2014. Reading mathematics for understanding—From novice to expert. *The Journal of Mathematical Behavior* 35 (2014).
- [58] Arjun Srinivasan, Steven Drucker, Alex Endert, and John Stasko. 2019. Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication. *IEEE Transactions on Visualization and Computer Graphics* 25 (2019).
- [59] Edward Tufte. 1985. *The visual display of quantitative information*. Graphics Press.
- [60] Bret Victor. *Tangle: a JavaScript library for reactive documents*. <http://worrydream.com/Tangle/> Last accessed August 15, 2021.
- [61] Theresa E. Wege, Sophie Batchelor, Matthew Inglis, Honali Mistry, and Dirk Schlamm. 2020. Iconicity in Mathematical Notation: Commutativity and Symmetry. *Journal of Numerical Cognition* 6 (2020).
- [62] Katherine Ye, Wode Ni, Max Krieger, Dor Ma'ayan, Jenna Wise, Jonathan Aldrich, Joshua Sunshine, and Keenan Crane. 2020. Penrose: from mathematical notation to beautiful diagrams. *ACM Transactions on Graphics (TOG)* 39 (2020).
- [63] Hsin I. Yung and Fred Paas. 2015. Effects of Computer-Based Visual Representation on Mathematics Learning and Cognitive Load. *J. Educ. Technol. Soc.* 18 (2015).
- [64] Robert C. Zeleznik, Andrew Bragdon, Ferdi Adeputra, and Hsu-Sheng Ko. 2010. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In *Proceedings of the Symposium on User Interface Software and Technology*.

A APPENDIX

A.1 Documents

What follows on the next two pages are two tables containing the IDs, types, titles, and public URLs of all documents analyzed in the content analysis (Section 3). Readers can also find additional information about the content analysis, including an annotation guide and a spreadsheet containing the complete analysis results for all 281 formulas, in the supplemental material.

ID	Type	Title	URL (if document is public)
P1	Research paper	Towards Grounding of Formulae	https://www.aclweb.org/anthology/2020.sdp-1.16.pdf
P2	Research paper	Continuous Prefetch for Interactive Data Applications	https://arxiv.org/pdf/2007.07858.pdf
P3	Research paper	Gamut: A Design Probe to Understand How Data Scientists Understand Machine Learning Models	https://fredhohman.com/papers/19-gamut-chi.pdf
S1	Research slides	On Unbalanced Optimal Transport: An Analysis of Sinkhorn Algorithm	https://icml.cc/virtual/2020/poster/6163
S2	Research slides	Neural Clustering Processes	https://icml.cc/virtual/2020/poster/6425
S3	Research slides	Deep reconstruction of strange attractors from time series	https://nips.cc/virtual/2020/protected/poster_021bbc7ee20b71134d53e20206bd6feb.html
S4	Research slides	Accountable Off-Policy Evaluation With Kernel Bellman Statistics	https://icml.cc/virtual/2020/poster/6683
S5	Research slides	Autoregressive Score Matching	https://nips.cc/virtual/2020/protected/poster_4a4526b1ec301744aba9526d78fcb2a6.html
S6	Research slides	Topological Autoencoders	https://icml.cc/virtual/2020/poster/5851
S7	Research slides	Meta-Consolidation for Continual Learning	https://nips.cc/virtual/2020/protected/poster_a5585a4d4b12277fee5cad0880611bc6.html
S8	Research slides	Deep Transformation-Invariant Clustering	https://nips.cc/virtual/2020/protected/poster_5a5eab21ca2a8fef4af5e35709ecca15.html
S9	Research slides	Neural Anisotropy Directions	https://nips.cc/virtual/2020/protected/poster_cff02a74da64d145a4aed3a577a106ab.html
S10	Research slides	Bayesian Deep Ensembles via the Neural Tangent Kernel	https://nips.cc/virtual/2020/protected/poster_0b1ec366924b26fc98fa7b71a9c249cf.html
D1	Distill	Why Momentum Really Works	https://distill.pub/2017/momentum/
D2	Distill	The Paths Perspective on Value Learning	https://distill.pub/2019/paths-perspective-on-value-learning/
W1	Web article	Pi (π) Explained Visually	https://setosa.io/ev/pi/
W2	Web article	Conditional Probability Explained Visually	https://setosa.io/ev/conditional-probability/
W3	Web article	Pixels and their neighbors: Finite volume	https://row1.ca/pixels-and-their-neighbors
W4	Web article	An Interactive Guide To The Fourier Transform	https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/
W5	Web article	Understanding the Fourier Transform	https://web.archive.org/web/20120418231513/www.altdevblogaday.com/2011/05/17/understanding-the-fourier-transform/
W6	Web article	Surprising Uses of the Pythagorean Theorem	https://betterexplained.com/articles/surprising-uses-of-the-pythagorean-theorem/
W7	Web article	An Intuitive (and Short) Explanation of Bayes Theorem	https://betterexplained.com/articles/an-intuitive-and-short-explanation-of-bayes-theorem/
W8	Web article	Image Kernels	https://setosa.io/ev/image-kernels/
W9	Web article	Differential Equations	https://observablehq.com/@hzsteinberg/differential-equations

Table 2: Documents analyzed in Study 1 (continued in Table 3). Each document is listed with the ID that identifies it in this paper, its type, title, and a URL at which the document can be accessed, if a public URL is available.

ID	Type	Title	URL (if document is public)
W10	Web article	Importance of Distance Metrics in Machine Learning Modelling	https://towardsdatascience.com/importance-of-distance-metrics-in-machine-learning-modelling-e51395ffe60d
W11	Web article	Wallpaper Generator - Volumetric light scattering, Part 1 of 2	https://chuckleplant.github.io/2017/05/28/light-shafts.html
W12	Web article	Rocket Golf	https://what-if.xkcd.com/85/
W13	Web article	Building a Powerful DQN in TensorFlow 2.0 (explanation & tutorial)	https://medium.com/analytics-vidhya/building-a-powerful-dqn-in-tensorflow-2-0-explanation-tutorial-d48ea8f3177a
W14	Web article	Build Your Own Tensorflow	https://taliesin.ai/projects/edu/indaba-2019/
W15	Web article	Food Discovery with Uber Eats: Using Graph Learning to Power Recommendations	https://eng.uber.com/uber-eats-graph-learning/
V1	Video	Derivation of the Navier-Stokes Equations	https://www.youtube.com/watch?v=zWdnf3Uh1RE
V2	Video	But what is the Fourier Transform? A visual introduction.	https://www.youtube.com/watch?v=spUNpyF58BY
B1	Book chapter	In pursuit of the unknown: 17 equations that changed the world (<i>Chapter 3 only</i>)	(no public URL)
B2	Book chapter	Everyday Data Science (<i>Chapter 7 only</i>)	(no public URL)
B3	Book chapter	Byrne's edition of Euclid's Geometry (<i>Book 1, first five propositions only</i>)	https://www.c82.net/euclid/book1/
B4	Book chapter	A Student's Guide to Maxwell's Equations (<i>Chapter 1 only</i>)	(no public URL)
B5	Book chapter	Inside Deep Learning (<i>Chapter 2 only</i>)	(no public URL)
B6	Book chapter	Human-in-the-Loop Machine Learning (<i>Chapter 3 only</i>)	(no public URL)
N1	Course notes	Exponential Improvements	http://www.andrusia.com/math/complex-numbers/eulers-identity.pdf
L1	Course slides	Diff-in-diff I & II	https://evalf20.classes.andrewheiss.com/slides/08-slides.pdf
T1	Tweet	Tweet by @redblobgames	https://twitter.com/redblobgames/status/1111699048472305664
T2	Tweet	Tweet by @mdekstrand	https://twitter.com/mdekstrand/status/1373400822470504451
T3	Tweet	Tweet by @kwinkunks	https://twitter.com/kwinkunks/status/1233142784217034757
O1	Other	Math and Analogies	https://betterexplained.com/articles/math-and-analogies/
O2	Other	Answer to "Visual Pythagorean demonstration"	https://matheducators.stackexchange.com/a/40
O3	Other	MathType: Tips to use with Microsoft Word	https://docs.wiris.com/en/mathtype/office_tools/tips/tips_word#drawing_attention_to_your_equations_with_comments_and_annotations
O4	Other	Example - Sine Waves in Time or Space	https://airladon.github.io/FigureOne/examples/Traveling%20Wave%2002%20-%20Sine%20Waves/

Table 3: Documents analyzed in Study 1 (continued from Table 2). Each document is listed with the ID that identifies it in this paper, its type, title, and a URL at which the document can be accessed, if a public URL is available.