



NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2025-40

November, 2025

Towards Practice-Integrated Human-AI Systems for Developing High- Quality Educational Assessments

Yuan “Charles” Cui

Abstract

Assessments are foundational to education, shaping how teachers teach, how students learn, and how accountability is maintained across the educational ecosystem. Yet, creating and delivering high-quality assessments remains one of the most time-consuming aspects of teaching. Educators, especially those in under-resourced schools, often lack the time, training, and support to design effective assessments that are aligned with curricula and diverse student needs. While recent advances in large language models (LLMs) show the potential to reduce this burden, they also raise concerns about quality and teacher trust. In specific domains such as data visualization education, researchers and educators face similar challenges: existing assessments for measuring visualization literacy (the ability to read and interpret visualizations) are lengthy, rigid, and difficult to scale or adapt. Across both general and visualization-specific contexts, improving efficiency, maintaining quality, and building trust in assessment development and delivery requires systems that lower technical barriers, promote content reuse and adaptation, and integrate meaningfully into educators’ existing workflows.

This dissertation addresses these challenges through three complementary projects that develop adaptive, scalable, quality-oriented solutions for assessment development. The first project

introduces A-VLAT and A-CALVI, computerized adaptive tests for visualization literacy that maintain precision and reliability while halving test length. The second project presents the VILA pipeline that leverages LLMs to generate visualization literacy questions at scale, combining automated generation with expert-driven evaluation to ensure quality and validity. The third project extends these ideas to general K–12 education, where we codesigned a web-based system, Ripplet, that supports multilevel reusable interactions with LLMs for creating, editing, and adapting assessments. Together, these projects contribute to the development of assessment systems that improve efficiency, scalability, and quality.

Keywords

Human-AI Collaboration, AI for Education, Data Visualization, Assessments

NORTHWESTERN UNIVERSITY

Towards Practice-Integrated Human-AI Systems
for Developing High-Quality Educational Assessments

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Yuan “Charles” Cui

EVANSTON, ILLINOIS

October 2025

© Copyright by Yuan “Charles” Cui, 2025

All Rights Reserved

Abstract

Assessments are foundational to education, shaping how teachers teach, how students learn, and how accountability is maintained across the educational ecosystem. Yet, creating and delivering high-quality assessments remains one of the most time-consuming aspects of teaching. Educators, especially those in under-resourced schools, often lack the time, training, and support to design effective assessments that are aligned with curricula and diverse student needs. While recent advances in large language models (LLMs) show the potential to reduce this burden, they also raise concerns about quality and teacher trust. In specific domains such as data visualization education, researchers and educators face similar challenges: existing assessments for measuring visualization literacy (the ability of read and interpret visualizations) are lengthy, rigid, and difficult to scale or adapt. Across both general and visualization-specific contexts, improving efficiency, maintaining quality, and building trust in assessment development and delivery requires systems that lower technical barriers, promote content reuse and adaptation, and integrate meaningfully into educators' existing workflows.

This dissertation addresses these challenges through three complementary projects that develop adaptive, scalable, quality-oriented solutions for assessment development and delivery. The first project introduces A-VLAT and A-CALVI, computerized adaptive tests for visualization literacy that maintain precision and reliability while halving test length. The second project presents VILA, a pipeline that leverages LLMs to generate visualization literacy questions at scale, combining automated generation with expert-driven evaluation to ensure quality and validity. The third project extends these ideas to general K-12 education, where we codesigned a web-based system, Ripplet, that supports multilevel reusable interactions with LLMs for creating, editing, and adapting assessments. Together, these projects contribute to the development of assessment systems that improve efficiency, scalability, and quality.

Acknowledgment

It took a village.

First, I would like to thank my advisor, Matthew Kay. I started my Ph.D. as a theory student but realized that I wanted to do more applied work midway through. It was a challenging time for me both as a student and as an early twenty-something who felt lost emotionally. I seriously considered leaving the program. Just on the verge of quitting the Ph.D., I took a seminar with Matt. He seemed kind, so I asked if I could work with him, and he took a chance on me. I had much to learn, and Matt has been patient, constructive, and always, always encouraging when teaching me the craft of research. Over the years, he has always made clear that he wants what makes me happy, whether it is a research project unrelated to his interest or a career outside of academia. I feel extremely fortunate to have an advisor who is invested in helping me find my path.

Fumeng Yang has been my de facto co-advisor. She was a postdoc in my lab when we first met. She had no obligation in mentoring me, but she was always generous with her time and helped me above and beyond. We became close collaborators over time, and she is my go-to person when I needed counsel. After she became a professor at the University of Maryland, she supported me to pursue a passion project as a visiting student in her lab. She taught me to be programmatic, resilient, and detail-oriented. Her impeccable aesthetic taste has also had a profound impact on me: research is not only about the science, but also about the art of presenting the science beautifully.

Lily Ge is my closest student collaborator. After I shifted my research focus, Lily let me work on a project with her. That was the first time in grad school when I felt joy doing research. That project turned out to be a success, and it inspired the subsequent projects in my dissertation. Lily is meticulous and humble, and she always held me accountable.

We challenged each other's ideas all the time and became better researchers. Ph.D. can be a lonely journey, and I am glad to have her as a companion on this road.

I am grateful for my collaborators in my dissertation projects. Lane Harrison and Yiren Ding at Worcester Polytechnic Institute provided crucial support for two projects, and they have offered me research and career advice outside of those projects. For my last thesis project, I worked with a team of brilliant undergraduate students at Northwestern, including Annabel Goldman, Jovy Zhou, Xiaolin Liu, Clarissa Shieh, Joshua Yao, Mia Cheng, April Shi, Laura Félix, Christopher Heo, Rachel Johnson, and Eric Lee. These students have taught me so much more than I could teach them.

I have been fortunate to have other support systems at Northwestern. The MU Collective Lab is my base camp, and its members (Mandi Cai, Sheng Long, Abhraneel Sarma, Ziyang Guo, Taewook Kim, Maryam Hedayati, Paula Kayongo, and Hyeok Kim) created a loving and supportive environment. The Northwestern Institute on Complex Systems and HCI + D Center have generously provided funding for my research, where I also met my thesis committee members Eleanor O'Rourke and Michael Horn, who have given me invaluable advice. I have also received guidance from many other professors, including Steve Franconeri, Jason Hartline, Haoqi Zhang, Konstantin Makarychev, Aravindan Vijayaraghavan, Elizabeth Tipton, and Duri Long. The student leaders at CSPAC and CSSI have built a strong sense of community within the computer science department, and my life has been better because of it. Elysse Longiotti at the Northwestern Career Development Center has always encouraged me. My work would not have been possible without the incredible staff members of the CS department, including Julia Blend, Katie Winters, Bella Barrios, Melissa Duong, Jensen Smith, Ethan Walles, and Dru Redmond.

I am lucky to have had opportunities to explore my other research interests during

my Ph.D. I want to thank Lily Xu, Ana-Andreea Stoica, Sandro Radovanović, Matthew Olckers, Francisco J. Marmolejo Cossio, Sera Linardi, Wanyi Dai Li, and Rediet Abebe at EAAMO (previously Mechanism Design for Social Good), Daniel Ho and Derek Ouyang at Stanford RegLab, Ugofilippo Basellini, Irena Chen, and Monica Alexander at the Max Planck Institute for Demographic Research, and Wei Zhang, Yiru Chen, Jane Hoffswell, and Abhisek Trivedi at Adobe. I am especially grateful for the Data Science for Social Good Fellowship program (Rayid Ghani, Kit Rodolfa, my teammates, and all the mentors and fellows) at Carnegie Mellon University—it was everything I could have hoped for in a truly collaborative, supportive, and intellectually stimulating environment. These experiences shaped me as a researcher, and more importantly, I made life-long friends who continue to help me grow as a person.

Many other people helped during my Ph.D. My undergraduate mentor Samuel Taggart at Oberlin College, a former Northwestern CS Ph.D. graduate, continued to support via numerous email replies and Zoom calls. I am lucky to have met people in the data visualization and HCI community who were willing to spend their time helping me, including Leilani Battle, Steven Moore, Katy Williams, Alex Kale, Niklas Elmqvist, Mi Feng, Cindy Xiong, Xiaoying Pu, and Bum Chul Kwon.

This journey would not have been possible without friends and family. The past five years were made so much better thanks to the kindness and companionship of Jennah Gosciak, Zeyuan Zhang, Xinnan Cheng, Daniel Firebanks-Quevedo, Tao Hong, Martin Mancini, Yuhao Wu, Kennedy Casey, Jane Hsieh, Brian Wesley Hill, Yasmine Ramachandra, Catherine Wahlenmayer, and the MatSci basketball teams. I am especially grateful for Hannah Sommerlad and the Sommerlad family for their support. 最后，感谢爸爸妈妈无条件的爱，也感谢他们从我十五岁起就让我自由地追求自己的人生。

Table of Contents

Abstract	3
Acknowledgment	4
Table of Contents	7
1 Introduction	10
1.1 Background and Motivation	10
1.2 Thesis Contributions	13
2 Related Work	16
2.1 Assessment Development	16
2.2 Automatic Question Generation	17
2.3 Human-AI Collaboration for Assessment Development	18
2.4 Computerized Adaptive Assessment	19
3 Adaptive Assessment of Visualization Literacy	21
3.1 Introduction	21
3.2 Background	24
3.3 Item Analysis	27

3.4	Construction of A-VLAT and A-CALVI Algorithms	29
3.5	Validity Evaluation	43
3.6	Reliability Evaluation	47
3.7	Discussion	51
4	VILA: Scalable Generation of Visualization Questions with LLMs	59
4.1	Introduction	59
4.2	Background	62
4.3	Item Generation	65
4.4	Item Evaluation	76
4.5	Application Demonstration: VILA-VLAT	84
4.6	Discussion	88
5	Ripplet: Authoring Educational Assessments through Multilevel Reusable Interactions with LLMs	93
5.1	Introduction	93
5.2	Background	97
5.3	Codesign Overview	98
5.4	Codesign Phase I: Conceptual Model and Design Objectives	101
5.5	Codesign Phase II: System Description of Ripplet	110
5.6	Codesign Phase III: Independent Use and Longitudinal Observations	123
5.7	Controlled User Study	127
5.8	Discussion	133
6	Discussion	136
6.1	Iterative Authoring of Questions with Visual Representations Remains Difficult	136

6.2	Lowering Technical Barriers Is a Prerequisite for Adoption and Impact . .	137
6.3	Mitigating Uneven Risks of AI Reliance Needs Carefully Designed Support	138
6.4	Designing Impactful Assessment Systems Requires the Participation of Ex- perts and Educators	140
6.5	Bridging Research and Practice Through Integrated Assessment Platforms	141
7	Conclusion	143
	References	144

Chapter 1

Introduction

1.1 Background and Motivation

Assessments—defined in this dissertation as paper- or computer-based instruments composed of answerable questions such as homework, quizzes, tests, and exams, rather than broader tasks like labs or class projects—are critical in the educational ecosystem. They affect how teachers teach and how students learn and ensure accountability in all state-holders [20, 124, 126]. Formative assessments such as quizzes and homework help teachers monitor students’ progress and guide instruction, while summative assessments such as unit tests and final exams provide a snapshot of student achievement for schools, parents, and policymakers [117, 130, 11]. Despite their central role, **creating and delivering high-quality assessments remains one of the most challenging and time-consuming responsibilities in education**. Teachers spend up to half their time writing and revising assessments, aligning them with standards, and adapting them for students with different needs [126, 140]. This intensive labor leaves little time for reflection or improvement, and it is compounded by systemic disparities: teachers in under-resourced schools often have fewer planning periods and limited access to high-quality materials. As a result, assessment quality and equity vary widely across classrooms [78, 124, 131, 45].

Efforts to improve the efficiency of creating and delivering high-quality assessments have a long history in educational research. Early computational methods sought to reduce teacher workload through template-based automatic question generation (AQG), where question developers encoded reusable question templates that could be programmatically instantiated with different parameters [37, 99, 78, 105, 134]. These approaches found some success in domains such as mathematics and medical licensing, but they often produced questions that were repetitive or lacked pedagogical depth. In recent years, advances in large language models (LLMs) have renewed interest in automating question creation [77, 65, 87, 31, 98]. LLM-based systems can generate multiple-choice, short-answer, or open-ended questions across a wide range of subjects. Yet these systems introduce new risks for quality and trust. LLMs may generate factually incorrect or ambiguous questions or overlook alignment with local curricula and classroom realities. More importantly, most existing tools treat teachers as passive consumers of AI output rather than as active collaborators. Assessment authoring is a personal and iterative process, grounded in teachers' pedagogical values and their nuanced understanding of student needs [37, 149, 84, 2]. Without mechanisms that preserve teacher agency and enable oversight, automation may produce large volumes of content that are efficient but unreliable—or worse, erode teachers' confidence in the assessment process itself.

Similar issues arise in specialized domains such as data visualization education, where assessing people's ability to interpret and reason about data visualizations—known as *visualization literacy*—is increasingly important [15, 81, 55]. Visualization literacy affects decision-making in high-stakes contexts ranging from public health to environmental policy, yet measuring this ability reliably and efficiently remains difficult. Existing assessments such as the Visualization Literacy Assessment Test (VLAT) [81] and the Critical Thinking Assessment for Literacy in Visualizations (CALVI) [55] provide strong founda-

tions but share many of the same limitations seen in general education: they are time-consuming to administer, rigid in structure, and labor-intensive to create, adapt, and maintain.

Across both general and visualization-specific contexts, a common theme emerges: assessments are vital but difficult to create, adapt, and deliver efficiently while maintaining quality and trust. Through the course of this dissertation, I explore these challenges and, in doing so, identify a set of design principles for addressing them. Across three projects, we find that effective assessment systems must (1) lower technical barriers for educators and researchers, (2) promote reuse and adaptation of existing high-quality materials, (3) integrate seamlessly into teachers' authentic workflows, and (4) preserve the transparency and control that underpin teachers' trust in AI-assisted processes. In K-12 classrooms, this means developing tools that empower teachers to collaborate with AI while retaining authority over generated content. In data visualization education, it means creating adaptive assessments efficient in measurement and automated methods to generate visualization questions that maintain quality. Together, these findings motivate my **thesis statement**:

Assessments in K-12 classrooms and data visualization education are important but difficult for educators to create, adapt, and deliver. Designing systems that lower technical barriers, reuse and adapt existing content, and integrate into teachers' existing practices can help improve the efficiency of assessment development and ensure trust and quality.

1.2 Thesis Contributions

This dissertation responds to these challenges in K-12 educational assessments and visualization literacy assessments. Across three projects, I develop solutions that make assessments adaptive in measurement, scalable in content generation, and integrated into teachers' authoring practices.

1.2.1 Adaptive Assessment of Visualization Literacy (Chapter 3)

This work is done in collaboration with Lily Ge, Yiren Ding, Fumeng Yang, Lane Harrison, and Matthew Kay.

To improve the efficiency of assessment delivery in data visualization education, we developed computerized adaptive tests (CATs) for visualization literacy. Building on existing assessments, including VLAT and CALVI, we applied item response theory and adaptive testing algorithms to create A-VLAT and A-CALVI, which measure the same skills in about half the number of questions. Specifically, we (1) employed item response theory (IRT) and non-psychometric constraints to construct adaptive versions of the assessments, (2) finalized the configurations of adaptation through simulation, (3) refined the composition of test items (questions) of A-CALVI via a qualitative study, and (4) demonstrated the test-retest reliability (ICC: 0.98 and 0.98) and convergent validity (correlation: 0.81 and 0.66) of both CATs via four online studies. Although these results demonstrate that adaptive assessments can more efficiently and reliably measure visualization literacy, their adoption in practice has been limited. Despite frequent citations, few researchers have deployed these adaptive assessments, largely because hosting the adaptive algorithms requires setting up a backend service, which is an obstacle even for visualization

researchers. Moreover, because existing question banks are small, repeated administrations to the same participants often reintroduce identical questions, limiting longitudinal use. These challenges reveal the need to lower technical barriers and to develop scalable methods for generating high-quality visualization assessment questions.

1.2.2 VILA: Scalable Generation of Visualization Questions (Chapter 4)

This work is done in collaboration with Lily Ge, Yiren Ding, Lane Harrison, Fumeng Yang, and Matthew Kay.

Building on the need of having large question banks from the previous project, we built VILA (**V**isualization **I**tems generated by **L**arge language models), a three-staged pipeline that uses large language models to generate visualization questions at scale. Working with visualization experts, we developed an evaluation rulebook that formalizes quality criteria and guided the review of over a thousand generated items (questions). The resulting VILA bank provides a validated collection of visualization questions. Through our evaluation, we found that 21% of generated questions contained errors, which we classified to reveal common pitfalls in LLM-generated assessment content for visualization. This analysis highlighted that maintaining question quality requires human oversight not only at the end of the generation process but throughout it. Human feedback can guide generation, correction, and validation iteratively. Despite the methodological contributions and frequent citations of VILA, adoption among visualization educators and researchers has remained limited, in part due to the technical barrier of setting up local Python environments and also the one-off nature of the generation workflow that is not naturally integrated into teachers' practices. These lessons reveal the need to design AI-based assessment systems that embed human oversight directly into the authoring process and fit into users' existing workflows.

1.2.3 Ripplet: Authoring Educational Assessments through Multilevel Reusable Interactions with LLMs (Chapter 5)

This work is done in collaboration with Annabel Goldman, Jovy Zhou, Xiaolin Liu, Clarissa Shieh, Joshua Yao, Mia Cheng, Matthew Kay, and Fumeng Yang.

Building on these lessons about the need for human oversight and better integration into educators' workflows, we develop Ripplet, a web-based and LLM-powered assessment authoring system. To ensure that the system can integrate into teachers' existing workflows, we partnered with 13 teachers to codesign Ripplet. Ripplet supports creating questions from diverse inputs, adapting content at multiple levels, and tracking and reapplying teachers' edits. Over seven months, Ripplet enabled teachers to create formative assessments they would not have otherwise made, shifted their practices from generation to curation, and helped them reflect more on assessment quality. In a user study with 15 additional teachers, compared to their current practices teachers felt the results were more worth their effort ($\mu = +1.93/10$, 95% CI: [0.33, 3.53]) and that assessment quality improved ($\mu = +1.11$ [0.10, 2.11]). Unlike VILA, Ripplet achieved sustained and organic adoption. The system is used by teachers across more than ten schools in four states and continues to attract new users through word of mouth, even after formal studies concluded. This reflects the importance of codesign in ensuring that systems fit into teachers' workflows. Rather than using LLMs to generate everything from scratch, Ripplet's design emphasizes reusing and adapting existing high-quality materials, allowing teachers to maintain ownership while benefiting from AI assistance. These outcomes demonstrate the thesis of this dissertation: systems which lower technical barriers, promote reuse and adaptation, and integrate seamlessly into teachers' authentic workflows can make assessment development more efficient and ensure assessment quality.

Chapter 2

Related Work

2.1 Assessment Development

Developing assessments is a time-intensive process [94, 140, 78, 124], consuming up to half of teachers' professional time [126, 125]. They need to align assessments with learning objectives, devise intellectually demanding tasks, and ensure instructional relevance [133, 56, 42]. However, most teachers receive little training or support for assessment development [124, 125, 131, 56, 86, 42, 95, 41, 131], leading to low-quality assessments and consequently uneven chances among students to demonstrate their knowledge [78, 124, 131, 45].

Researchers in education, psychology, and cognitive science have proposed measurement theories and models to guide assessment development, such as constructing detailed test blueprints [35, 136] and estimating item parameters (e.g., difficulty) [5, 50, 64]. Some argue that models of cognition and learning can inform educational assessment development, making assessments more effective in measuring student understanding [107]. However, translating such research into practice is not easy [94, 107]. In reality, teachers rarely adopt such principles when creating assessments, as these theories often

lack relevance to their day-to-day instruction [94, 128]. For instance, most theories of educational measurement are designed for large-scale or standardized testing, emphasizing psychometric properties rather than the formative and summative purposes of classroom assessment. As a result, these theories often feel disconnected from teachers’ day-to-day realities, leading them to rely on experience and institutional norms instead of abstract theoretical principles. Consequently, existing work in test development theory provides insufficient practical guidance for how teachers currently develop assessments.

2.2 Automatic Question Generation

Questions are the building blocks of an assessment. Traditional question creation processes are highly costly and laborious due to their manual nature. They usually involve subject-matter experts writing and reviewing questions. Thus, researchers have sought ways to reduce the amount of human labor required in this process. This field of study is *automatic question generation* (AQG) [79]. AQG methods aim to ease the burden of creating questions [37, 99, 78], gaining momentum with the recent advances in generative AI [99, 78]. These methods range from template-based generation to neural and transformer models [105, 134, 148, 135, 113, 23, 141]. Recent LLM-based methods dramatically lowered the technical barrier to entry [77, 65, 87, 31, 98, 49, 46], and can take inputs like Bloom’s Taxonomy and textbooks [90, 39, 89, 88] to generate questions of varying difficulty levels [26, 30] in multiple choice, free response, and fill-in-the-blank formats [99]. While these methods have shown promise in generating questions across diverse contexts, they often struggle to produce high-quality questions that can measure higher cognitive abilities [90, 39, 53, 30, 49].

Additionally, there exists a gap between these automated methods and teachers’

practices. Most automated solutions are not designed for teachers: some are end-to-end pipelines that require technical expertise (e.g., coding) to execute, while others use generic interfaces that lack targeted support for assessment authoring. In addition, effective assessments must closely align with individual teachers’ curricula, institutional guidelines, and the unique needs of student cohorts [106, 41, 46]—requirements that cannot be met by sifting through thousands of generic questions. Assessment authoring is also a deeply personal practice that embodies teachers’ pedagogical philosophy, beliefs, and years of accumulated knowledge about their students [37, 149, 84, 2]. Automated methods without careful design to involve human oversight do not fit real-world classrooms.

2.3 Human-AI Collaboration for Assessment Development

Prior work has explored ways to incorporate human oversight into LLM-based question generation [85, 73, 31, 48, 120]. TutorCraftEase allows teachers to generate questions with LLMs and accept, reject, and manually edit LLM outputs. This enables teachers to produce quality questions more efficiently [73]. Similarly, ReadingQuizMaker helps college instructors make questions for reading quizzes, providing control for when to request AI assistance; the authors found that instructors preferred this collaborative approach over an AI-only process [85]. While these systems show promise in mitigating the issues of fully automated approaches, they operate largely at the level of individual questions and provide limited support for reusing and adapting questions. Assessment authoring is inherently **multilevel**: it requires not only generating individual questions but also structuring them into a holistic assessment, tweaking specific parts of a question to fit students’ needs, balancing difficulty and topic coverage, and aligning the collection to curricular goals. In addition, as we find in Chapter 5, assessment authoring is also **it-**

erative: teachers move between creating and adapting questions, searching for relevant materials, reviewing and restructuring the content, and refining their requirements for an assessment. Designers of assessment development technology should consider the multilevel and iterative properties of this process.

2.4 Computerized Adaptive Assessment

After developing the content of an assessment, educators often face the challenge of assessment delivery. Traditional assessment delivery methods often rely on fixed tests that treat all students the same, regardless of their individual ability levels. These assessments can be lengthy and inefficient. Moreover, they can increase the risk of disengagement or even cheating, as students face repetitive or poorly matched questions that fail to adapt to their skill level [121].

Computerized adaptive testing (CAT) is a form of computer-based assessment where the next question or the next set of questions a test-taker sees depends on their performance on the previous questions. The idea is to adaptively select questions to tailor to a test-taker's ability in order to gain more information about them: for example, if someone performs poorly on difficult questions, they will then be presented with an easier question. CAT has many advantages over traditional static assessments where everyone receives the same questions: it can achieve similar measurement precision with fewer questions, and it may better motivate test takers because the questions are more appropriate for their ability level [121].

CAT has been broadly adopted in healthcare research. Clinical researchers have developed various CATs to measure patient characteristics, such as quality of life [58], anxi-

ety [60, 138], and mental health disorders [59]. CAT has also been used to measure various forms of literacy, such as English literacy [51], health literacy [72], and math knowledge of university students [57]. CAT has also been widely applied to many large-scale standardized tests in practice, such as the Graduate Record Examinations (GRE) [47] and the Graduate Management Admission Test (GMAT) [122], and continues to grow in popularity: the SAT, a long-established college admissions test in the U.S., became digital and adopted adaptive testing since 2023 and 2024 [36]. Despite the prevalent prior applications and potential benefits of CAT, it has yet to be applied in data visualization education.

Chapter 3

Adaptive Assessment of Visualization Literacy

This work is done in collaboration with Lily Ge, Yiren Ding, Fumeng Yang, Lane Harrison, and Matthew Kay. This chapter is published as a full paper at the IEEE VIS Conference in 2023 [38].

3.1 Introduction

Visualization literacy—an individual’s ability to understand and interpret visualizations—can significantly impact data-driven decisions. People may rely on data visualizations showing the spread of infectious diseases to make personal health decisions, to decide between treatment options in medical settings, to make financial decisions, or to engage with social or political topics. Inaccurate interpretations of such visualizations may lead to faulty reasoning and decisions, as well as harmful outcomes.

In the study of visualization literacy, there is a persistent need for accurate, reliable, and timely ways to assess people’s ability. For example, visualization researchers may want to track the progress of this ability over time, or to empirically evaluate interventions designed to enhance a target group’s ability to interpret visualizations. Such efforts require concise, quantitative assessments of visualization literacy that can be adminis-

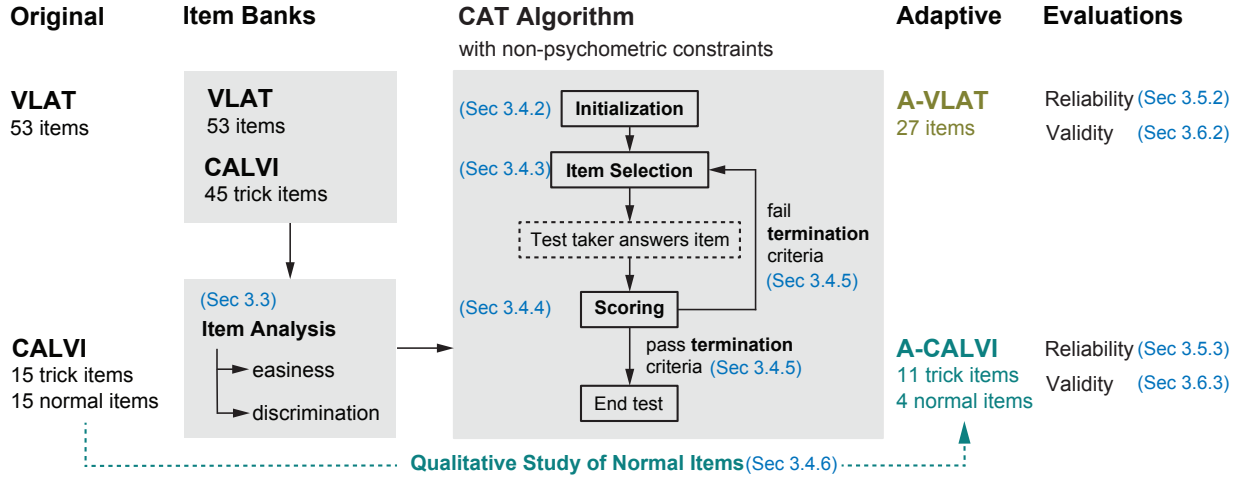


Figure 3.1: The process of developing adaptive visualization literacy assessments: **A-VLAT** and **A-CALVI**. We start with the item banks (i.e., question banks) of VLAT and CALVI and use the parameters from item analysis to construct the CAT algorithms for the adaptive assessments. We then evaluate their validity and reliability via four online studies. The annotations in blue are the components' corresponding sections.

tered to individuals multiple times, allowing for the measurement of skill development and the evaluation of intervention efficacy through pre- and post-testing.

While researchers have devised assessments to measure both basic visualization literacy skills [81, 15] and the critical thinking ability to detect visualization misinformation [55], these tests can be time-consuming. To address this, we apply *computerized adaptive testing* (CAT) to visualization literacy. CAT's core principle is to adaptively select test items (i.e., questions)¹ for a test taker based on their performance on previously-answered items, and it can provide a precise estimate of the test-taker's abilities with fewer items. CAT has been widely applied in healthcare science [58, 60, 138, 59] and used in practice by educational agencies, such as the Educational Testing Service (ETS) and College Board, to create large-scale standardized tests [47, 122, 36]. However, visualization literacy assessments have yet to take advantage of the benefits of adaptive testing.

¹In this dissertation, the terms *item* and *question* are used interchangeably.

In this project, we adopt the CAT development framework [132] to create two short, adaptive visualization literacy tests: **A-VLAT** and **A-CALVI**, which are built upon the existing static assessments VLAT [81] and CALVI [55]. First, we compute *item parameters* (how easy items are and how well items in the bank separate test takers of different abilities) with Item Response Theory (IRT). Using the item parameters, we construct adaptive algorithms to select items for test-takers. We include non-psychometric constraints in our algorithms to ensure the tests contain balanced content; i.e., to ensure **A-VLAT** covers all 12 chart types and 8 tasks from VLAT and **A-CALVI** covers all 11 misleaders from CALVI. We also conduct a qualitative study to refine the composition of items in **A-CALVI**, reducing its length by a further 11 items. Ultimately, we contribute:

1. A demonstration of the refinement of visualization literacy assessments through adaptive testing, including the incorporation of non-psychometric features of existing assessments.
2. Two valid and reliable adaptive visualization literacy tests, **A-VLAT** (27 items) and **A-CALVI** (15 items), which are half the length of their non-adaptive counterparts.
3. Evidence from four online studies demonstrating the test-retest reliability (ICC: **0.98** and **0.98**) and convergent validity (correlation: **0.81** and **0.66**) of these tests.

In addition, we discuss how cumulative results from visualization literacy studies can better inform our understanding of the relationship between the constructs measured by literacy assessments (e.g., what correlations between chart types might tell us about how people understand visualizations). We also provide recommendations for using and customizing visualization literacy assessments based on test administrators' needs. Our work just scratches the surface of the potential for adaptive testing in visual-

ization; we believe it offers a path to shorter, repeatable, and more reliable assessment of visualization literacy.

3.2 Background

3.2.1 Visualization Literacy

Within the visualization community, many researchers have studied visualization literacy through the development of assessments [15, 81, 55] and frameworks [14]. Boy et al. used Item Response Theory (IRT) to generate a set of tests that aims to assess people’s ability to interpret line charts, bar charts, and scatterplots [15]. Similarly, to test people’s ability to interpret visually represented data, Lee et al. developed a Visualization Literacy Assessment Test (VLAT) that contains 53 multiple-choice items [81]. Later, considering the complexity of visualization literacy as a construct, Ge et al. expanded on prior definitions to incorporate the ability to identify and reason about visualization misinformation and developed a Critical Thinking Assessment for Literacy in Visualizations (CALVI), which has a bank of 45 items [55]. We rely heavily on both VLAT and CALVI to develop our adaptive tests, **A-VLAT** and **A-CALVI**, so we explain VLAT and CALVI in more detail below.

VLAT. The items in VLAT were generated from 12 chart types with 8 tasks such as *retrieve value* and *make comparisons*, and each test taker is expected to take all 53 items [81]. For example, Figure 3.2.A is a VLAT item in a pie chart that asks viewers to make comparisons. Each item in VLAT has an item difficulty and item discrimination index obtained from Classical Test Theory (CTT) analysis [81].

CALVI. The items in CALVI were generated from 11 misleaders (i.e., ways a chart can lead

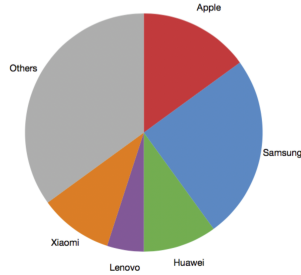
to conclusions not supported by the data) and 9 chart types [55]. CALVI consists of 15 *trick* items, which are items whose visualizations contain a misleader, and 15 *normal* items (i.e., items based on well-formed visualizations)—CALVI includes normal items because people may encounter misleading visualizations mixed in with well-formed visualizations in their daily lives. It is worth noting that only test takers' performance on the trick items are used to estimate their abilities to detect visualization misinformation. The 15 trick items are selected to cover all 11 misleaders from CALVI's bank of 45 total trick items [55]. For example, Figure 3.2.B is a trick item with misleader *Manipulation of Scales - Inappropriate Use of Scale Functions*, where the number labels on the y-axis do not match the spacings of the tick marks; and Figure 3.2.C is a normal item. The trick items in CALVI each have an item easiness and an item discrimination parameter estimated using a 2-parameter IRT model [55].

Besides the assessments above, related fields such as statistical literacy also investigated graph interpretation abilities, and measurements were developed to assess graph comprehension. Specifically, studying how students interpreted graphical representations of distributions, DelMas et al. developed assessment items for the ability to interpret distributions [40]. Other researchers have also developed instruments that target younger audiences [108], and some focused more on the ability to critically interpret graphs, which is related to statistical literacy [4].

Although existing assessments provide measurements for visualization literacy and related abilities, we reiterate that they can be time consuming and inflexible due to their fixed, lengthy format. Next, we describe the procedure that we adapted for developing computerized adaptive tests, which reduced the lengths of these tests while maintaining similar measurement precision.

The global smartphone market share of Apple is larger than that of Huawei.

Global Smartphone Market Share (%)

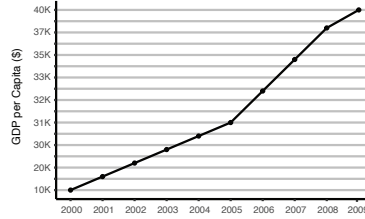


True
False
Skip

(A)

True or False:
GDP per capita for country Z grew faster from 2000 to 2003 than from 2005 to 2008.

Annual GDP per Capita for Country Z

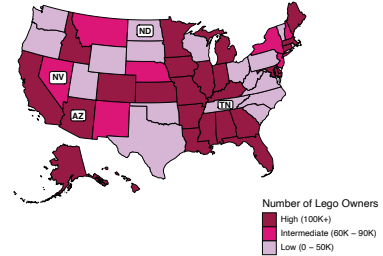


True
False
Cannot be inferred / inadequate Information

(B)

Which of the following states has the highest number of Lego owners?

Number of Lego Owners



AZ
NV
TN
Cannot be inferred / inadequate Information

(C)

Figure 3.2: Example items from VLAT and CALVI: (A) is an item from VLAT; (B) is a trick item from CALVI; (C) is a normal item from CALVI.

3.2.2 Procedure of CAT Development

Although existing measurement methods are not adaptive, as explained in Section 3.2.1, they provide a valuable foundation for the development of CAT, which requires item parameters from IRT item analysis. We extend and adopt the framework proposed by Thompson and Weiss [132] to develop adaptive tests. Below, we outline steps from this framework that were instrumental to developing our test.

First, test developers need to conduct **item analysis** of the item bank using test tryout data (i.e., data collected on participants answering items before the test is deployed in practice). In this stage, IRT item analysis is applied on test tryout data to compute the item easiness (i.e., how easy an item is) and item discrimination (i.e., how well an item differentiates test takers of different levels of ability) parameters of the items in the bank. After the test developers obtain these parameters for the items, they can proceed to the

construction of CAT algorithms. A CAT algorithm consists of four main components:

1. Initialization: initialize an estimate of the test-taker's ability;
2. Item selection: select the next item for the test-taker based on their current estimated ability (score);
3. Scoring: compute the score for a test-taker after each item;
4. Termination: end the test when the termination criterion is met.

To build a CAT algorithm, test developers use the test tryout data, the item parameters, and simulation studies to determine the configurations of these main components and demonstrate the measurement precision of the CAT. Although the framework thus far does not outline steps specifically for establishing reliability and validity, in the development of tests in general, they should be assessed [35]. Therefore, for both CATs we also evaluate the **reliability** (i.e., examining the extent to which the test produces consistent and stable results) and **validity** (i.e., ensuring that the test measures the ability that it is supposed to measure) [35].

3.3 Item Analysis

As outlined in Section 3.2.2, the first stage of CAT development is item analysis. In this section, we describe the IRT model used for item analysis to obtain parameters for both item easiness and discrimination. These item parameters are necessary for constructing the CAT algorithms, to be detailed in the following sections. Additional results from the item parameter analysis are available in the OSF repository of this project.

3.3.1 2-Parameter IRT Model

IRT mathematically describes the relationship of a test taker's ability, item parameters, and the probability of them answering a particular item correctly. We use the 2-parameter IRT model to infer the item easiness and discrimination parameters of all VLAT and CALVI items. The item response function of the 2-parameter IRT model (Equation (3.1)) describes the relationship between the probability of a test taker with ability θ answering item i correctly (left-hand side) and the easiness b_i and discrimination a_i of item i (right-hand side). This function will be used in conjunction with test tryout data to compute the item parameters (i.e., a_i and b_i) for VLAT and CALVI items, and it will also be used in Section 3.4 to define and compute other key quantities (e.g., item information, standard error) in adaptive testing.

$$p_i(\theta) = \frac{1}{1 + \exp(-a_i(\theta + b_i))} \quad (3.1)$$

3.3.2 Parameters of VLAT and CALVI Items

The developers of CALVI collected test tryout data and used the 2-parameter Bayesian IRT model to compute the item parameters of each item [55]. Therefore, we can directly use these item parameters to adaptively select the next item for [A-CALVI](#).

Because item analysis for VLAT was conducted with CTT instead of IRT, we apply IRT analysis on VLAT items using their test tryout data.² We use a similar 2-parameter Bayesian IRT model as the one used for CALVI with adjustments to address the following difference between the two tests: CALVI has two types of items: normal items and trick

²This data was obtained from the authors of VLAT by personal communication.

items, and only the trick items are associated with test takers' abilities θ . Whereas in VLAT, a person's performance on every item counts. All other specifications are the same as the model from CALVI, provided in the OSF repository, along with the item parameters of the VLAT and CALVI items.

3.4 Construction of A-VLAT and A-CALVI Algorithms

To construct A-VLAT and A-CALVI, we use test tryout data, item parameters, and simulation studies to determine the configurations of A-VLAT and A-CALVI: **(1) initialization** (Section 3.4.2), **(2) item selection** (Section 3.4.3), **(3) scoring** (Section 3.4.4), and **(4) termination** (Section 3.4.5). In addition, we conduct a qualitative study for A-CALVI to determine how to balance items across the misleading / normal categories (Section 3.4.6). Figure 3.3 presents these components and the flow of our CAT algorithm.

3.4.1 Key Quantities in IRT-based CAT

The idea of CAT is to iteratively give a test taker an item that is best suited to their ability and iteratively update their estimated ability. In IRT-based CAT, an item that is best suited to a test taker with ability θ is the item that provides the most information about that test taker; this quantity is called *item information*. Equation (3.2) provides the mathematical definition of item information in the 2-parameter model [6]: on the left-hand side, $I_i(\theta)$ is the item information from test taker with ability θ answering item i ; on the right-hand side, a_i is the item discrimination parameter of item i , and $p_i(\theta)$ is the probability of person with ability θ answering item i correctly (defined in Equation (3.1)).

$$I_i(\theta) = a_i^2 p_i(\theta)(1 - p_i(\theta)) \quad (3.2)$$

Notice that $I_i(\theta)$ is maximized when $p_i(\theta) = 0.5$. This means that an item provides the most information about a test taker with ability θ when we are most uncertain about whether the test taker will get this item right.³ Also, the greater the item discrimination a_i (an item's ability to differentiate test takers of different levels of ability), the greater the item information.

Under the context of item information, the item that is best suited to a test taker with ability θ is the item with the highest information for θ in the item bank. This will be used in item selection (Section 3.4.3).

Given the definition of item information, we can also define the *test information* [6] as the sum of the item information of its items:

$$I(\theta) = \sum I_i(\theta) \quad (3.3)$$

Another use of item information is to quantify the measurement precision of the test. The *standard error* [6] is defined as

$$SE(\theta) = \frac{1}{\sqrt{I(\theta)}}. \quad (3.4)$$

The larger the test information, the smaller the standard error, and the more precise the measurement of the test on ability θ . The standard error will be used in the simulation studies to construct a metric for comparing the measurement precision of our adaptive tests with their original static counterparts, and this metric will provide evidence for selecting appropriate lengths for the adaptive tests (Section 3.4.5).

³This should not be confused with the test taker being most uncertain about which answer is correct (i.e. randomly picking an option). For instance, if a test taker picks randomly in a 3-option item, $P(\text{correct})$ would be $1/3$ and $P(\text{incorrect})$ would be $2/3$ —so we would be more certain that they would answer incorrectly than correctly. Uncertainty is maximized at $1/2$, not $1/3$.

3.4.2 Initialization

As shown in Figure 3.3, the first step of the CAT algorithm is to initialize θ for a test taker, which can then be used for item selection. After examining the test tryout data of both VLAT and CALVI, we find that the distributions of θ are approximately normal for both tests. Thus, most people's θ 's are close to the mean of the distribution. Therefore, we initialize the θ of both adaptive tests using the mean of the θ 's in respective test tryout data. We explore the implications and possibilities of different methods for initializing θ in the discussion section.

3.4.3 Item Selection

The next item in **A-VLAT** or **A-CALVI** will be the item that yields the highest item information for the test taker's current θ estimate. However, simply selecting without constraints could risk not covering the important aspects of the original test, which is a salient trade-off of shorter adaptive tests. This is referred to as *content balancing* [123, 137].

To ensure content balancing, we impose non-psychometric constraints to the item selection process such that **A-VLAT** covers all 12 chart types and 8 tasks of VLAT and **A-CALVI** covers all 11 misleaders of CALVI. We refer to them as *non-psychometric features*.⁴

This is achieved by the following logic: we initialize the list of non-psychometric features to contain all features needed to be covered. If the remaining number of items in the test is **greater than the length of this list**, then the item with the highest information is selected from the bank, and its associated non-psychometric features are removed from the list. If the remaining number of items in the test is **equal to the length of this list**, the

⁴The list of non-psychometric features is provided in the OSF repository.

highest-information item with at least one of the remaining features is selected, and its associated non-psychometric features are removed from the list.

3.4.4 Scoring

In adaptive testing, the θ (ability) estimate needs to be updated every time a test taker answers an item (shown in Figure 3.3). We initialize θ to a prior distribution based on the test tryout data from participants taking VLAT and CALVI. Then, with each answer, we perform a Bayesian update to adjust the distribution of the θ estimate. We compute the final score of a test taker by taking the mean of the distribution of the θ estimate at the end of this iterative process.

3.4.5 Termination

We develop a fixed-length CAT for **A-VLAT** and **A-CALVI**. While CATs can be variable-length or fixed-length, variable-length tests can be difficult to implement in formal learning spaces with fixed time constraints, and require additional considerations and procedures to develop. We determine the lengths using simulations⁵ by comparing the standard error of each person in fixed-length CAT of various lengths to that of the original static version of the test. In Section 3.4.5.1 and Section 3.4.5.2, we present details of the simulation studies resulting in **27 items for A-VLAT** and **11 trick items for A-CALVI**.

Because we select the original static tests as the baseline for both **A-VLAT** and **A-CALVI**, we define the following metric:

$$\text{relative difference in } SE(\theta) = \frac{SE_A(\theta) - SE_O(\theta)}{SE_O(\theta)}, \quad (3.5)$$

⁵The code and data for simulations can be found in the OSF repository.

where $SE_A(\theta)$ is the standard error for θ in the adaptive test and $SE_O(\theta)$ is the standard error for θ in the original test. This metric is a measure of relative precision of the adaptive test compared to the original version; the smaller relative difference in $SE(\theta)$ is, the more precise the adaptive test compared to the original version. For example, a relative difference in $SE(\theta)$ of 0.30 implies that for this θ , the standard error of the adaptive test is within 30% of the standard error of the baseline. Similarly, a relative difference in $SE(\theta)$ of 0 means that the adaptive and original versions are equally precise. Note that if this quantity is positive, the adaptive test has a larger standard error than the baseline for θ . If the quantity is negative, then the adaptive test has a smaller standard error than the baseline for θ .

3.4.5.1 Simulation for A-VLAT

In this simulation study, we compare the measurement precision of A-VLAT with VLAT and provide reasoning for why we select 27 items as the length for A-VLAT.

Data. The item analysis of VLAT test tryout data on 191 participants yielded θ estimates for all 191 participants. We fit a normal distribution for θ using this data and generate a sample of 500 simulated persons (i.e., θ 's). We treat the simulated θ 's as the ground truth latent abilities for the simulated persons and refer to them as **true θ 's** below.

Method. Because our goal is to find a reasonable fixed length for A-VLAT, we build a simulation that realizes the 500 simulated persons' test results under A-VLAT of different lengths from 19 items⁶ to 53 items and the original version of VLAT, which has a length of 53. We do so by (1) plugging in the true θ and the item parameters into Equation (3.1)

⁶The non-psychometric constraint of A-VLAT dictates that it must cover all 12 chart types and 8 tasks. Since any item is associated with a chart type and a task, A-VLAT needs to have at least 19 items to satisfy the non-psychometric constraint in the worst case.

to compute the probability of correctness for each simulated person, (2) simulating the binary correctness outcome with that probability, and (3) computing the mean of the posterior distribution of the θ estimate as the final score for each person as described in Section 3.4.4.

From here, we compute the relative difference in $SE(\theta)$ for all 500 simulated persons.

Results. Figure 3.4 shows the 500 simulated persons' distribution of relative difference in $SE(\theta)$ for **A-VLAT** of length 19 to 53. Upon inspecting the trend, we find that this chart has approximately four segments: relative difference in $SE(\theta)$ experienced the sharpest drop from length 19 to length 26, a milder decrease from 27 to 36, another steep yet short downfall from 37 to 40, and finally plateaus after 41. Note that it never goes below 0 because the baseline uses all 53 items, and adaptively selecting 53 items would yield the same result.

After examining Figure 3.4, we select 27 as the length of **A-VLAT**. This reduces the length of the original test by half, and the median and most observations of relative difference in $SE(\theta)$ are below 0.15, a reasonable loss of measurement precision in exchange for halving the test. It should be noted that there is no gold standard for selecting the perfect cutoff in the literature. Test administrators can use Figure 3.4 to help customize the length based on their needs and constraints. We will discuss customization of our adaptive tests in Section 3.7.3.

3.4.5.2 Simulation for **A-CALVI**

In this simulation study, we compare the measurement precision of **A-CALVI** with CALVI and provide reasoning for why we select 11 as the number of trick items for **A-CALVI**.

Data. Similar to the **A-VLAT** simulation, we used the θ estimates for all 497 participants from CALVI test tryout to generate a sample of 500 simulated persons. We treat all simulated θ 's as the ground truth latent abilities for the simulated persons.

Method. The same method in Section 3.4.5.1 was applied. we obtained the relative difference in $SE(\theta)$ for all simulated persons.⁷

Results. Figure 3.5 shows a downward trend. From length of 13, the median and most observations of relative difference in $SE(\theta)$ fall below 0, indicating that **A-CALVI** outperforms the original CALVI in terms of measurement precision; this is because **A-CALVI** selects better items from the bank of 45 than the fixed 15 items in CALVI. We select 11 as the length of **A-CALVI** because it is shortest length that can cover all non-psychometric features and the median and most of the observations of relative difference in $SE(\theta)$ at length 11 are below 0.10.

3.4.6 Composition of **A-CALVI** Items via Qualitative Analysis

The developers of CALVI included 15 items with well-constructed non-misleading visualizations (i.e. normal items) to accompany 15 items with misleading visualizations (i.e. trick items). This is because (1) if a test taker realizes that all items are about misleading visualizations, they could adopt the strategy of never selecting the obviously correct answers and (2) when people encounter visualizations in the real world, they see both normal, well-constructed and misleading visualizations, which makes distinguishing misleading ones difficult [55]. However, a test taker's performance on the 15 normal items does not affect the θ estimate in CALVI, and the normal items constitute half of the test,

⁷Similar to **A-VLAT**, **A-CALVI** needs at least 11 items to cover all 11 misleaders because each item in the bank is associated to only one misleader. The original CALVI has 15 trick items, so we selected 15 as an upper bound.

making it long and time-consuming. Therefore, we conduct a qualitative study to investigate the effect of normal items and whether they would be necessary in [A-CALVI](#).

We recruited two groups of participants from Prolific for two conditions: (1) in *without-normal condition*, participants took [A-CALVI](#) without normal items (i.e., only 11 adaptively-selected trick items from the bank of 45 trick items); (2) in *with-normal condition*, they took [A-CALVI](#) with normal items (i.e., 11 adaptively-selected trick items + 11 fixed normal items). The 11 normal items were randomly sampled from the set of 15 normal items in the item bank of CALVI.

After participants in both conditions completed the items on the test, we asked them to answer a post-test survey of Yes/No and open-ended questions⁸ to investigate the possible impact of the presence of normal items. These questions included whether they noticed items with misleading charts (i.e., trick items) and how they approached the rest of the test after noticing trick items. We also provided the participants with the items they saw and their responses (presented in the order in their test) to help them recall the test content. In the *with-normal* condition, we also explored how participants approached the rest of the test after noticing the presence of normal items.

Participants. A total of 60 participants were recruited from Prolific for this qualitative study, with 30 for the *without-normal condition* and 30 for the *with-normal condition*. The data of one participant in the *with-normal condition* was discarded due to a data collection error, leaving a total of 59 participants. All participants fall under the following criteria: speak fluent English, have normal or corrected-to-normal vision, are between 18 and 65 years old, and are U.S. residents. Participants who failed more than half of the attention check questions were excluded, and all participants passed the attention check in this

⁸The open-ended questions can be found in the OSF repository.

study. In addition, Prolific participants from similar studies that the authors conducted in the past were excluded.

Procedure. We started by presenting the consent form and information page describing the structure of the study to the participants. We instructed participants that they need to select an answer for the current item before proceeding to the next one, and once they moved on, they could not return to previous ones. After the participants completed the test items, they were directed to the post-test survey.

Method. We reviewed and coded participants' responses on the post-test survey to understand their experience with the test and how trick and normal items affected their approach to the test.

Results. In the *without-normal condition*, 24 out of 30 participants reported they noticed the presence of trick items, after which 16 paid more attention or became more skeptical of the items, 6 did not change how they approached the rest of the items, and 2 gave unclear responses.

In the *with-normal condition*, 19 out of 29 participants reported that they noticed the presence of trick items, after which 14 of them paid more attention or double checked their answers, 4 did not change how they approached the rest of the items, and 1 participant gave an unclear response. When asked whether the presence of normal items changed their approach to answer trick items, 16 out of the 19 said no, and the remaining 3 mentioned that it only made them more suspicious of the normal items.

There seems to be little evidence that the presence of normal items affected how participants approached the trick items, but we acknowledge that this sample size is small and the possibility of normal items having an effect still exists. Thus, we refrain from

completely removing normal items, but instead look to reduce the amount.

Our underlying goal is the same as CALVI—the test taker sees a mix of trick and normal items, and we reason about this from the the test takers’ perspective. Based on the analysis of trick items in the original CALVI, there are certain misleaders that are substantially harder to identify, such as *Overplotting* or *Missing Data* [55], meaning these items are essentially perceived as “normal” in the eye of the test taker. Therefore, the items from such misleader categories can be effectively considered as “normal” items by the test takers (i.e., perceived “normal”). We also notice that it is highly likely that the option of “Cannot be inferred/inadequate information” is the correct answer when it appears in a trick item in A-CALVI. To prevent the strategy of always selecting this option when it appears, we decide to include in A-CALVI all of the normal items from CALVI that contain this option but it is not the correct answer: there are 4 such items. With the 4 *actual* normal items and the *perceived* “normal” items, we expect that, from the test taker’s perspective, there would be a balanced mix of trick and “normal” items, following a similar format as the original CALVI. Thus, A-CALVI consists of 11 trick items selected adaptively by the CAT algorithm and 4 *actual* normal items. The total length of A-CALVI is 15, half of the length of the original CALVI.

3.4.7 Final Configurations of A-VLAT and A-CALVI

A-VLAT. The final configuration initializes the first item by using the mean of θ ’s from the VLAT test tryout data to select the item with highest information in the bank. After the test taker answers each item, that item is removed from the bank, and θ is updated accordingly, which is then used to select the next item. Each time, A-VLAT selects the next item with the highest information under the constraint that the 12 chart types and 8 tasks must be covered before the test terminates. A-VLAT terminates after a test taker has

answered 27 items.

A-CALVI. The final configuration initializes the first item using the mean of θ 's from the CALVI test tryout data to select the item with highest information in the bank. After the test taker answers each item, that item is removed from the bank, and θ is updated accordingly, which is then used to select the next item. Each time, **A-CALVI** selects the next item with the highest information under the constraint that the 11 misleaders must be covered before the test terminates. The 4 normal items are positioned at 4 randomly predetermined locations among the trick items, but which normal item appears at which of the 4 positions is randomly determined at runtime for each test taker. **A-CALVI** terminates after a test taker has answered 11 trick items and 4 normal items.

With the configurations of **A-VLAT** and **A-CALVI** finalized, it is necessary to know whether the two tests can provide stable and consistent measurements and whether they measure the abilities they are designed to measure. Therefore, we conduct four online studies to evaluate the reliability (Section 3.6) and validity (Section 3.5) of the adaptive tests.

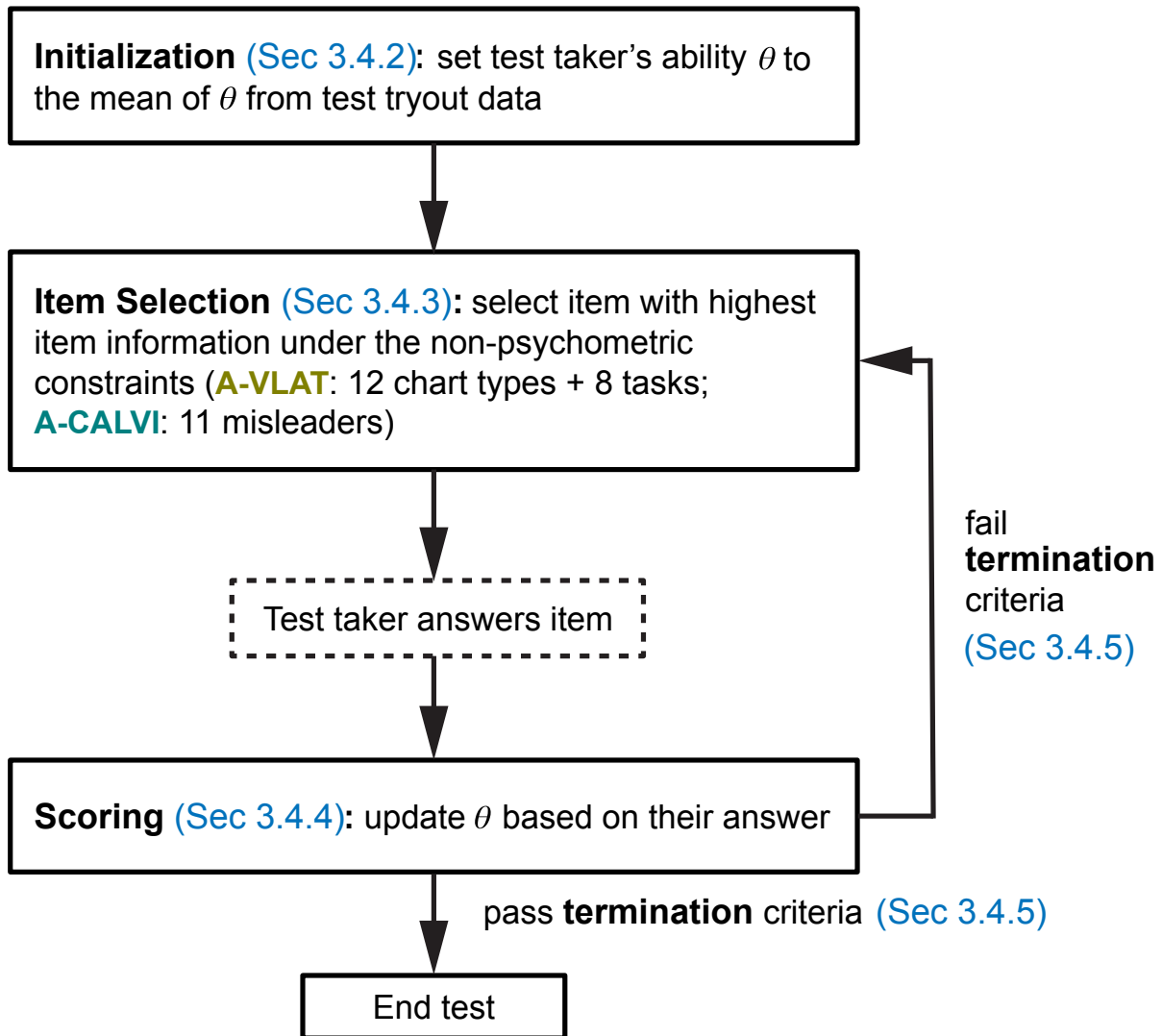


Figure 3.3: The components and flow of our CAT algorithm: the algorithm begins by initializing ability θ for a test taker, which is then used to select the next item. The test taker then answers the item, and their θ gets updated based on the answer. If the termination criterion is satisfied, then the test ends; if not, the updated θ is used to select the next item and this process repeats.

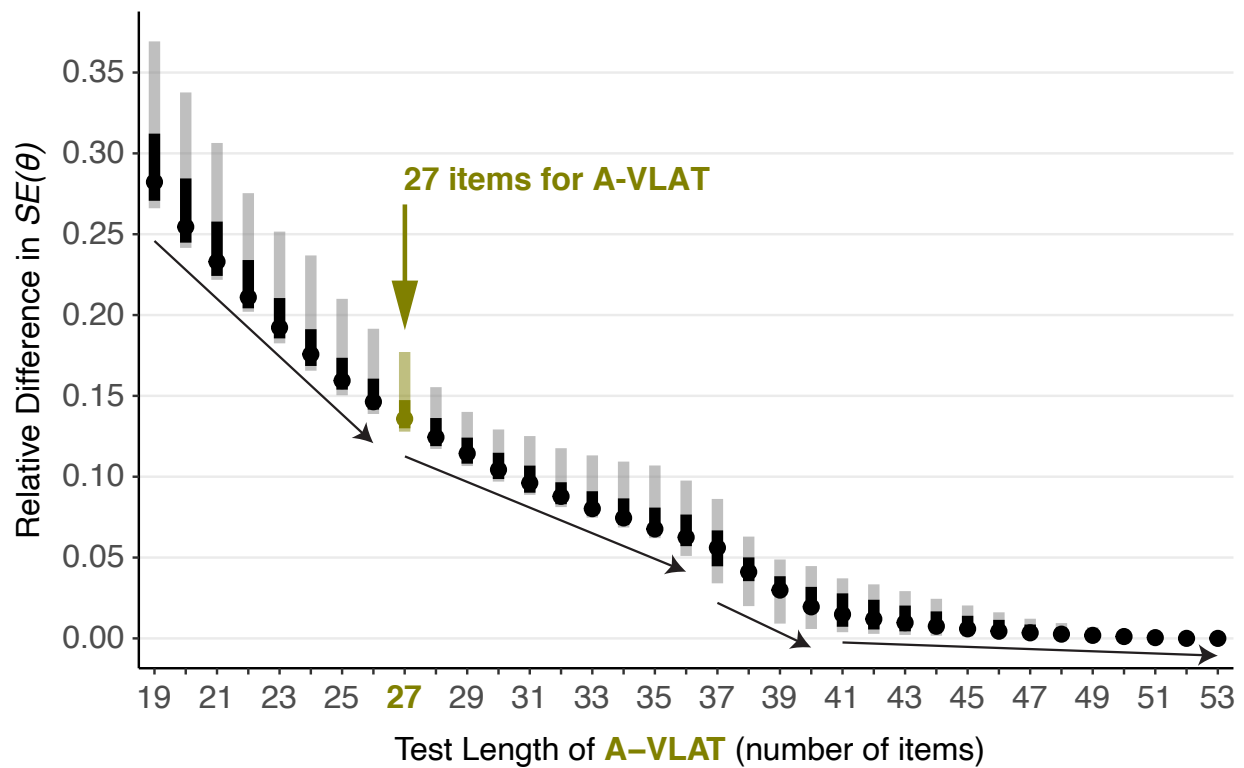


Figure 3.4: The distribution of relative difference in $SE(\theta)$ from length 19 to 53 for **A-VLAT**: the dot is the median, the lighter bar is where 95% of the observations fall, and the darker bar is where 66% of the observations fall. The trend has approximately 4 segments as indicated by the arrows. We select 27 as the length of **A-VLAT**.

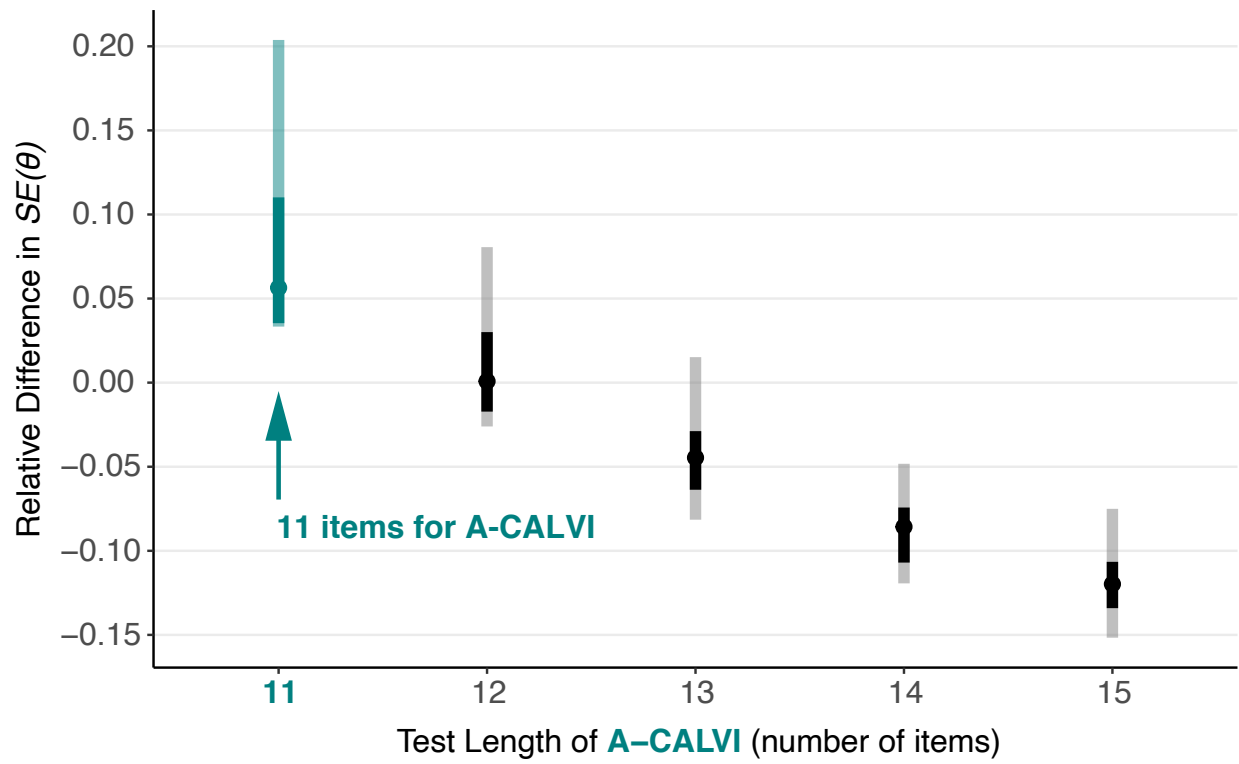


Figure 3.5: The distribution of relative difference in $SE(\theta)$ from length 11 to 15 for A-CALVI, which shows a downward trend. We select 11 as the length of A-CALVI.

3.5 Validity Evaluation

We evaluate whether our adaptive tests are valid—that they measure what they are designed to measure—by computing the **convergent validity** [33]: the correlation between test takers’ performance on the adaptive tests and their performance on the original static tests. This is an appropriate measure of the validity of the adaptive tests [62] because VLAT and CALVI are both valid measurements [55, 81] and the equivalence of the original and adaptive tests through strong correlation would demonstrate that the adaptive tests indeed measure the same abilities as the originals, hence are valid.

To compute the correlation coefficient for convergent validity, the same group of participants need to take both versions of the tests. Because retaking the tests may create learning and ordering effects, we use two groups of participants to counterbalance the effects. For both **A-VLAT** and **A-CALVI**, we have the same study design with two groups, both recruited from Prolific: each person takes both tests, with a week⁹ in between sessions. Order is counterbalanced: half the participants (Group 1) takes the original test first, and the other half (Group 2) takes the adaptive test first. The procedure of our validity studies is the same as that in Section 3.4.6.

3.5.1 Model for Validity Metric

Correlation coefficients can be computed via a Bayesian bivariate Normal model. We again use a measurement error model. The model is preregistered and can be found in

⁹Because of the similarity of content between the adaptive and original tests, we choose to have a one-week interval to mitigate learning effects.

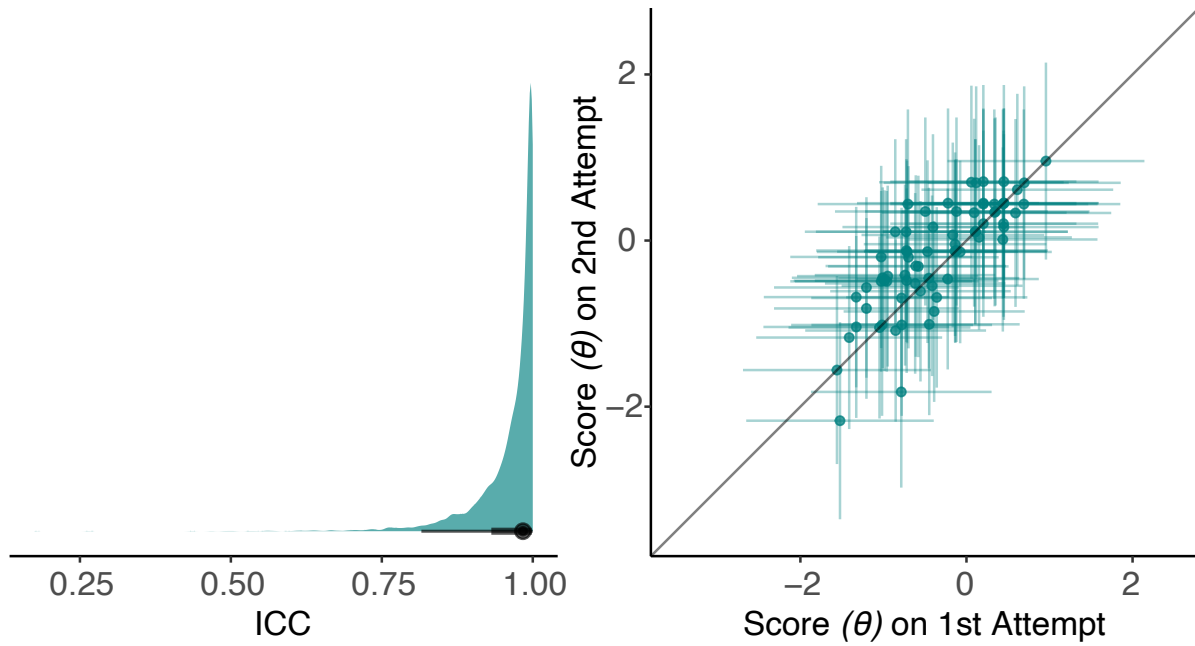


Figure 3.6: Reliability of [A-CALVI](#): the left plot is a posterior density plot of ICC; the right plot is a scatterplot of the 73 participants' scores on the first and second attempts with measurement error.

the OSF repository.¹⁰ The bivariate Normal model is:

$$\begin{bmatrix} \theta_0^* \\ \theta_A^* \end{bmatrix} \sim \text{Normal}_2 \left(\begin{bmatrix} \mu_0 \\ \mu_A \end{bmatrix}, \sigma_0, \sigma_A, \rho \right)$$

$$\theta_0 \sim \text{Normal}(\theta_0^*, \text{se}(\theta_0))$$

$$\theta_A \sim \text{Normal}(\theta_A^*, \text{se}(\theta_A)).$$

where θ_0^* and θ_A^* are the latent abilities measured from the original test and the adaptive test, with μ_0 and μ_A as their means, σ_0 and σ_A as their standard deviations, and ρ as the correlation between them. θ_0 and θ_A are the observed abilities, with measurement errors $\text{se}(\theta_0)$ and $\text{se}(\theta_A)$. In the implementation of this model, we re-parameterized μ_0 and μ_A in

¹⁰The model results in this section are based on an alternative implementation different than the initially pre-registered for the same reason in Section 3.6.1.

terms of the difference between them by subtracting the mean of θ_0 from all the observed θ 's.

For the prior of the model, we selected $\text{Normal}(0, 1)$ for the difference between θ_0 and θ_A , as 1 is a large difference in latent IRT ability. We picked $\text{Normal}(1, 0.5)$ for σ_0 and σ_A , because standard deviations should center around 1 as the IRT model is parametrized so that latent abilities have a standard deviation around 1. For ρ , we selected a uniform prior (the LKJ distribution with parameter $\eta = 1$).¹¹

We ran our final model with 4 chains, each with 20,000 iterations. We discarded 10,000 warmup iterations per chain and thinned the final sample by 5, yielding 8,000 total post-warmup draws.

This model outputs the distribution of the correlation coefficient ρ for convergent validity.

3.5.2 Validity of **A-VLAT**

Participants. 120 participants were recruited (60 for each group) in anticipation around 100 would return to the second part of the study so that we have a sample size close to 100. This sample size was determined by collecting pilot data with 10 participants in each group, fitting the model with pilot data, generating simulated participants with the fitted model, and choosing the sample size that would yield a ± 0.1 95% credible interval around ρ .

¹¹We pre-registered LKJ distribution with parameter $\eta = 3$ as the prior of ρ in the initial implementation of the model for **A-VLAT**, because it needed a tight prior to converge. However, the alternative implementation—which does not estimate all θ 's—converges much more easily, so we relaxed the prior on ρ .

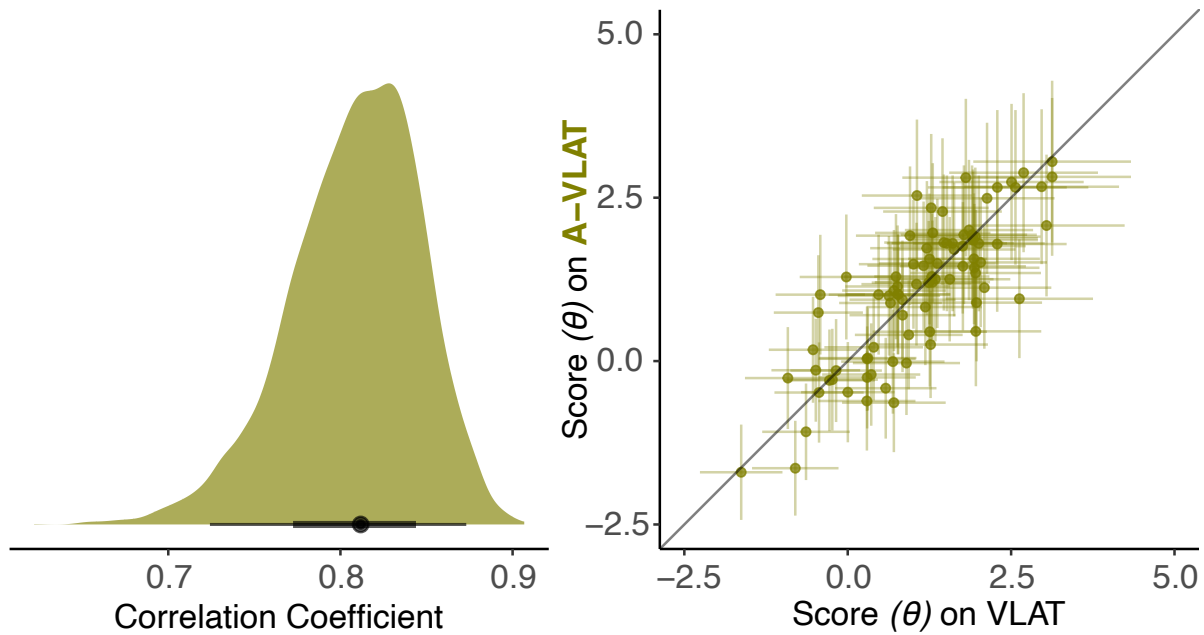


Figure 3.7: Validity of **A-VLAT**: the left plot is a posterior density plot of correlation coefficient ρ ; the right plot is a scatterplot of the 86 participants' scores on VLAT and **A-VLAT** with measurement error.

In the second part of the study, 49 returned in Group 1 and 37 returned in Group 2, for a total of 86 participants. All participants fall under the same criteria listed in Section 3.4.6.

Model Diagnostics. The minimal bulk effective and the minimum tail effective sample sizes are 7,607 and 7,394, and the $\hat{R}$ values are approximately 1.00 for all parameters.

Results. Figure 3.7 shows the posterior density of ρ and a scatterplot of participants' scores on the adaptive test versus those on the original test. **The median ρ is 0.81 with 95% CI of [0.72, 0.87].** While there is no gold standard for convergent validity, a correlation coefficient below 0.50 should be avoided and above 0.70 is recommended [25], so **A-VLAT** has **high convergent validity**.

3.5.3 Validity of A-CALVI

Participants. A sample size of 100 was determined in the same way as in Section 3.5.2. Based on the attrition rate of the previous study, 140 participants were recruited (70 in each group), in anticipation that around 100 participants would return to the second part of the study.

In the first part of the study, one participant's data was discarded because they failed the attention check questions. In the second part, 47 and 43 returned in Groups 1 and 2, for a total of 90 participants. All participants fall under the same criteria listed in Section 3.4.6.

Model Diagnostics. The minimal bulk effective and the minimum tail effective sample sizes are 6,455 and 5,090, and the $\hat{R}$ values are approximately 1.00 for all parameters.

Results. Figure 3.9 shows the posterior density of ρ and a scatterplot of participants' scores on the adaptive test versus those on the original test. **The median ρ is 0.66 with 95% CI of [0.53, 0.76].** Since a correlation coefficient below 0.50 should be avoided and above 0.70 is recommended [25], A-CALVI has **acceptable convergent validity**.

3.6 Reliability Evaluation

Reliability of a test is whether it can provide consistent and stable measurement. We provide evidence of **test-retest reliability**, a common way of assessing reliability and used in various domains to evaluate the reliability of CATs (e.g., [63, 9]).

Test-retest reliability evidence is gathered by administering a test to the same group of people twice with a time interval between the first and the second administrations and

comparing participant’s performances on each attempt. An interval of 1 or 2 weeks is typical [109]. For both **A-VLAT** and **A-CALVI**, we have the same study design: we recruit one group of participants from Prolific. Everyone takes the adaptive test and is invited back to take it again after a week.

We use the **intraclass correlation coefficient (ICC)** as the reliability metric recommended by literature [109]. Next, we describe the models to compute ICC for **A-VLAT** and **A-CALVI** and present the results.

3.6.1 Model for Reliability Metric

ICC can be computed via a random effects model, and we used a Bayesian measurement error version of the model because the measurement for each person is a distribution rather than a point estimate. It is important to take into account the measurement error when computing ICC, as otherwise it will be underestimated [146]. Our model is preregistered and can be found in the OSF repository.¹² The model is:

$$\begin{aligned}\theta^* &\sim \text{Normal}(\mu + \alpha_j, \sigma_\epsilon) \\ \theta &\sim \text{Normal}(\theta^*, \text{se}(\theta)) \\ \alpha_j &\sim \text{Normal}(0, \sigma_\alpha),\end{aligned}$$

where θ^* is the latent ability, μ is the unobserved mean, α_j is the random effect for person j , σ_ϵ is the standard deviation within-person, σ_α is the standard deviation between people, θ is the observed ability, and $\text{se}(\theta)$ is the measurement error of θ . For the priors

¹²We initially pre-registered an implementation of this model that is difficult to fit and computes extra parameters (the latent θ^* s) that we do not need. We later found an alternative parameterization better suited for our purpose, so the results in this section are from the alternative parameterization.

of this model, we selected $\text{Normal}(0, 1)$ and $\text{Normal}(-1, 1)$ for μ for **A-VLAT** and **A-CALVI**, respectively, because they are approximately the distributions of the estimated abilities in the VLAT and CALVI test tryout data. We also selected $\text{Normal}^+(0, 1)$ for σ_α and σ_ϵ .

For both the reliability of **A-VLAT** and **A-CALVI**, we ran our model with 4 chains, each with 20,000 iterations. We discarded 10,000 warmup iterations per chain and thinned the final sample by 5, yielding 8,000 total post-warmup draws.

This model outputs distributions of σ_α and σ_ϵ , which we then use to compute ICC with the following formula [83]:

$$\text{ICC} = \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_\epsilon^2}.$$

ICC can be understood as the ratio between the variance between people (σ_α^2) and the sum of variance between people and the (undesirable) within-person variance (σ_ϵ^2) [83].

3.6.2 Reliability of **A-VLAT**

Participants. 90 participants were recruited in anticipation that around 60 would return to the second study so that we have a sample size close to 60. This sample size was determined by collecting pilot data with 20 participants, fitting the model with pilot data, generating simulated participants with the fitted model, and choosing the sample size that would yield a ± 0.1 95% credible interval around the ICC.¹³

In the end, 47 participants completed both parts of the study. All participants fall under the same criteria listed in Section 3.4.6.

Model Diagnostics. The minimal bulk effective and the minimum tail effective sample

¹³All pilot analyses for sample size determination are in the OSF repository.

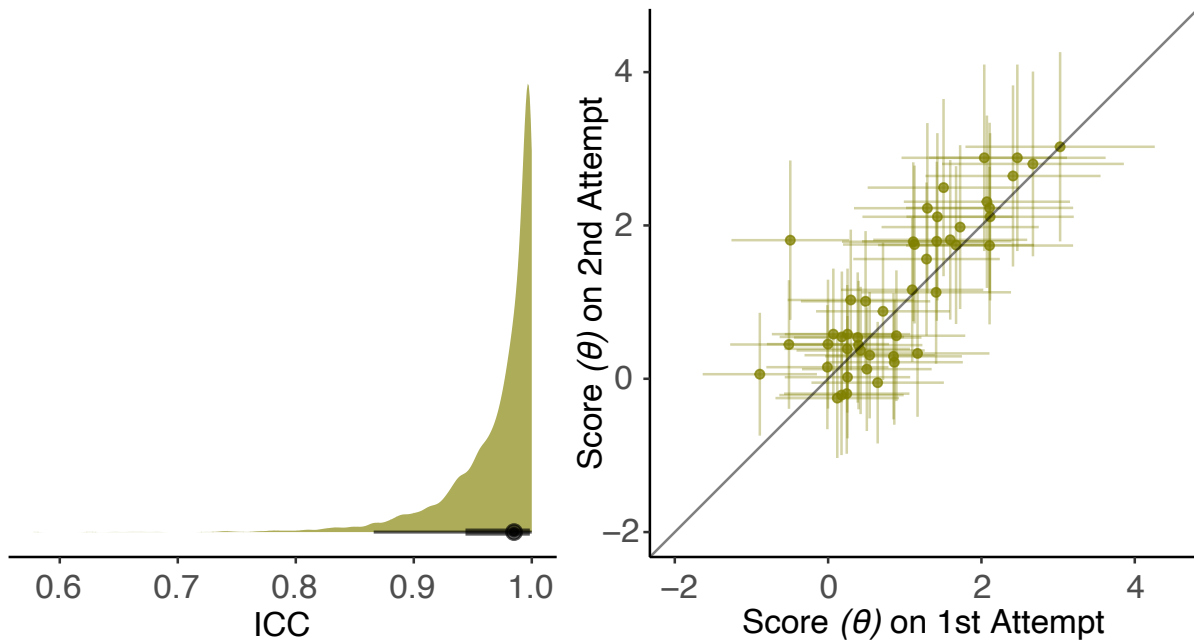


Figure 3.8: Reliability of **A-VLAT**: the left plot is a posterior density plot of ICC with a point interval plot at the bottom; the point is the median, and the lighter and darker bars are the 95% and 66% intervals. The right plot is a scatterplot of the 47 participants' scores on the first and second attempts, where each point represents a participant, and the cross lines at each point represents the measurement error (95% intervals).

sizes are 6,757 and 7,067, and the $\hat{R}$ values are approximately 1.00 for all parameters.

Results Figure 3.8 shows the posterior density of ICC and a scatterplot of participants' scores on the first attempt versus those on the second attempt. **The median ICC is 0.98 with 95% CI of [0.87, 1.00], so **A-VLAT** has excellent test-retest reliability [34, 93].**

3.6.3 Reliability of **A-CALVI**

Participants. We determined the sample size by first following the same method in Section 3.6.2. However, in the model with measurement error on pilot data, the intervals of ICC did not shrink as the sample size of the simulated participants increased. Therefore, we use a sample size of 60 because it is consistent with the sample size of the reliabil-

ity study for **A-VLAT** and the 95% intervals on σ_α and σ_ϵ have similar width as those in Section 3.6.2. In addition, we report the ICC results from models with and without measurement error because the interval of ICC may be large for the model with measurement error. Based on the attrition rate of the previous study, we recruited 115 participants, in anticipation that around 60 would return to the second study. In the end, 73 participants from the first part of the study returned to the second part. All participants fall under the same criteria listed in Section 3.4.6.

Model Diagnostics. The minimal bulk effective and the minimum tail effective sample sizes are 7,272 and 7,560, and the $\hat{R}$ values are approximately 1.00 for all parameters.

Results. Figure 3.6 shows the posterior density of ICC and a scatterplot of participants' scores on the first attempt versus those on the second attempt. **The median ICC is 0.98 with 95% CI of [0.82, 1.00]**, so **A-CALVI** has **excellent test-retest reliability** [34, 93]. Even though the model with measurement error is able to produce a small interval for ICC with data from this study, we report the result from the other model as stated in pre-registration: the median ICC is 0.77 with 95% CI of [0.66, 0.85]. We select the ICC from the model with measurement error as the final report for reliability because a model that does not account for measurement error consistently underestimates ICC. [146].

3.7 Discussion

3.7.1 How Reliable, Valid, and Precise Are CATs Really?

The results from Section 3.6 and Section 3.5 have shown that we have successfully created two reliable and valid adaptive tests, **A-VLAT** and **A-CALVI**, which measure the same abilities as the original static VLAT and CALVI. In addition, we collected data that can be

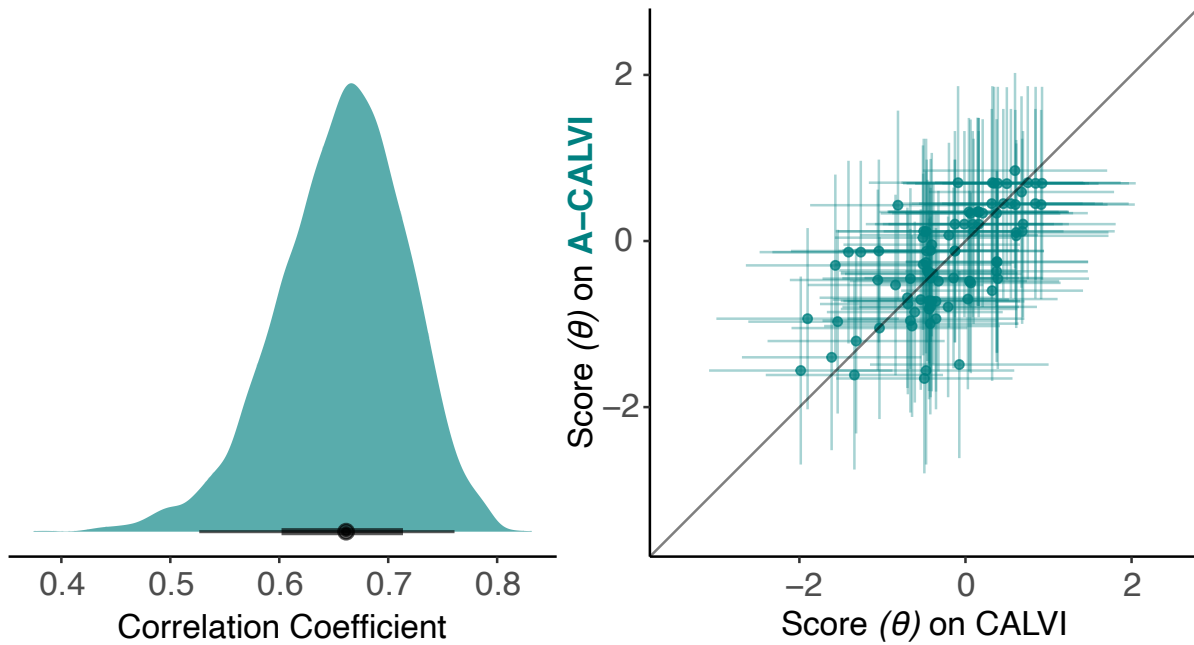


Figure 3.9: Validity of **A-CALVI**: the left plot is a posterior density plot of the correlation coefficient ρ ; the right plot is a scatterplot of the 90 participants' scores on CALVI and **A-CALVI** with measurement error.

used to demonstrate the measurement precision of both adaptive tests: in validity evaluation, every participant was asked to take both the adaptive and original versions. We can use this data to analyze the relative difference in $SE(\theta)$ of both adaptive tests as we did in Section 3.4.5 with simulation data.¹⁴

For **A-VLAT**, the distribution of relative difference in $SE(\theta)$ for 86 participants has a mean of 0.089 (i.e., for a person with average ability in this group, the standard error for them is within 8.9% of the standard error of the baseline.) with a standard deviation of 0.11.

For **A-CALVI**, the distribution of relative difference in $SE(\theta)$ for 90 participants has a mean of 0.023 with a standard deviation of 0.039.

¹⁴The code for this analysis is provided in the OSF repository.

We observe that results of relative difference in $SE(\theta)$ from actual participants reveal that both adaptive tests are similarly precise compared to their original static counterparts, in accordance with the simulation results in Section 3.4.5.1 and Section 3.4.5.2. This demonstrates the measurement precision of **A-VLAT** and **A-CALVI** and further supports the justification for the shortened lengths of the adaptive tests.

Another consideration of reliability is whether a test taker can recover quickly if they mistakenly select a wrong answer. First, we define d_i as the absolute value of the difference between the mean of the θ estimate distribution and the true θ of a test taker after answering the i -th item in the test. Then, “mistakenly selecting a wrong answer for item i ” is defined as when $d_i > d_{i-1}$; i.e., when their θ estimate moves away from the true θ after answering an item. Naturally, “recovery” from choosing a wrong answer for item i happens at the first step i' where $d_{i'} \leq d_i$; i.e., their θ estimate moves back to being within at least d_i of their true θ , and recovery length is $i' - i$. We conducted simulations¹⁵ to study the recovery lengths in our adaptive tests. We generated 500 simulated participants for each test as before, and simulated their response correctness to **A-VLAT** and **A-CALVI**. The median recovery lengths for **A-VLAT** and **A-CALVI** are 3 and 2 with standard deviations of 4.8 and 1.1, respectively. Therefore, test takers can recover quickly from a mistake in our tests, demonstrating good reliability.

3.7.2 Correlation within Chart Types, Tasks, and Misleadings

Items in the banks of VLAT and CALVI span a range of chart types, tasks, and misleadings. Studying the relationship between people’s performance within these different categories of visualization features can shed light on how different aspects and skills of visualization literacy connect and transfer, which can then help us better understand this ability and

¹⁵The code for the simulations is provided in the OSF repository.

potentially further optimize adaptive testing.

Actual participants have been asked to answer these visualization literacy items from prior work [55, 81], as well as the qualitative study, reliability evaluation, and validity evaluation in this project. This gives us the opportunity to use this data to examine the relationship between participants' performance on different chart types, tasks, and misleaders. Therefore, we conducted an exploratory study to compute the correlations within these visualization features by using both test tryout data (VLAT and CALVI) and data collected in this project and employing the same IRT models described in Section 3.3.¹⁶ Next, we present some highlights from this exploration and discuss their implications.

VLAT Chart Types. We found high correlations between performance on certain chart types: 0.83 with 95% CI of [0.72, 0.92] for scatterplot and bubble chart, 0.82 with 95% CI of [0.72, 0.90] for 100% stacked bar and stacked bar charts, and 0.81 with 95% CI of [0.69, 0.90] for scatterplot and line chart. There are also chart types without high correlations with others: choropleth map's and stacked area chart's correlation with other chart types do not exceed 0.38 and 0.52, respectively.

VLAT Tasks. Many pairs of tasks have high correlations: 0.93 with 95% CI of [0.88, 0.97] for *retrieve value* and *make comparisons*, 0.89 with 95% CI of [0.81, 0.96] for *find extremum* and *make comparisons*, 0.79 with 95% CI of [0.70, 0.86] for *determine range* and *retrieve value*. Most tasks highly correlate with at least another task.

CALVI Misleaders. Most pairs of misleaders have weak or moderate positive correlations: *Manipulation of Scales - Inappropriate Scale Range* and *Manipulation of Scales - Inappropriate*

¹⁶The entire results can be found in the OSF repository.

Use of Scale Functions have a correlation of 0.75 with 95% CI of [0.58, 0.89], highest among all pairs of misleaders. Some misleaders have very weak correlations with others; *Overplotting* especially stands out as its correlations with the other misleaders range from -0.21 to 0.34. This is reasonable as *Overplotting* is one of the most difficult misleaders to judge correctly.

Although **A-VLAT** and **A-CALVI** are significantly shorter than their original versions, there are opportunities to further shorten them by leveraging these correlations. For instance, the VLAT tasks *retrieve value* and *make comparisons* are highly correlated, so test developers could make the item selection process less constrained by only covering one of the two instead of both. Further research on incorporating these correlations into CAT algorithms for visualization literacy is needed.

Moreover, these correlations are valuable quantitative data that informs taxonomy in visualization literacy: people’s performance on tasks *retrieve value*, *find extremum*, and *make comparisons* are highly correlated, which means it may be reasonable to categorize them into one task. Additionally, when researchers and educators design curricula or tools to improve people’s visualization literacy, they can draw insights from these correlations to create more efficient and effective interventions: since people’s performances on answering items with scatterplots and bubble charts are strongly correlated, teachers may choose to focus on one, or group lessons together for increased efficiency.

3.7.3 Customization of Adaptive Tests

When using **A-VLAT** and **A-CALVI**, test administrators can customize many components of the tests, including initialization, item selection, and termination criterion. For initialization, we chose the means of the θ distributions from test tryouts. However, test

administrators can explore other alternatives, such as randomly choosing the first item from the bank to decrease the item exposure (i.e., the frequency at which an item is administered across all administrations) of certain items. Test administrators can also customize the non-psychometric constraints in item selection. For example, if one wishes to focus on only covering all VLAT chart types or only covering all tasks, they could adjust the non-psychometric constraints to achieve that. Moreover, the results from the simulation studies in Section 3.4.5 can guide administrators to adjust the length of the adaptive test based on their time-limit constraints and requirement for measurement precision. For scoring, we recommend reporting the mean of the posterior distribution of θ as a point estimate (described in Section 3.4.4), as well as the raw correctness (i.e., number of items answered correctly over the total number of items answered) to help test takers intuitively interpret their final scores.

Test administrators may wonder about the reliability and validity of their customized tests. We think that validity would remain relatively stable as the lengths of adaptive tests change so long as the non-psychometric features are covered, because these features ensure that the adaptive and original versions are qualitatively similar. Also, we think that the more non-psychometric features not covered, the less valid the adaptive tests would likely be. However, if two features are very similar, as discussed in Section 3.7.2, not covering one may not have a noticeable effect on validity. In addition, test-retest reliability could vary if the length is too short or long. On the one hand, if too short, the θ estimates would be more susceptible to measurement error and have limited variability, which can affect reliability. On the other hand, if a test is too long, test takers' motivation, attention, and effort might change on the second attempt, causing the test-retest reliability to decrease. Although test length can affect reliability and validity in extreme cases, we believe that moderately changing the length of **A-VLAT** or **A-CALVI** would have a marginal effect

on reliability and validity so long as the non-psychometric constraints are not violated.

For test administrators who only need to measure the visualization literacy of a group of people once, they can use **A-VLAT** and **A-CALVI** as alternatives to their original versions because they are much shorter, contain balanced content, and have similar measurement precision. For researchers and educators who want to conduct pre-post testing to evaluate the effectiveness of their interventions by administering a visualization literacy test once before and once after the intervention, we recommend using our adaptive tests for both administrations. For the second administration, they can choose to exclude certain overused items in the first attempt if they are concerned about item exposure. They could also use the estimated θ of a particular person in the first attempt to initialize this person's first item in the second attempt.

3.7.4 Limitation and Applicability in Practice

To create assessments under the IRT framework, it is necessary to construct items, run a test tryout study, and compute item parameters using IRT models. This process requires substantial resources as well as technical expertise on IRT modeling, which can be a barrier to implementing adaptive tests in practice. Moreover, adaptive assessments require more complex infrastructure to deliver than a standard series of questions and it is not trivial to add newly designed test items, due to the adaptive nature requiring computation at each step. However, given the benefits of CATs, there is a need to democratize community access to capabilities for test development and administration. Both **A-VLAT** and **A-CALVI** are therefore implemented using an experimental framework, designed to create flexible and reusable visualization assessments. One key feature is the decoupling of static and dynamic experiment/assessment content, which supports the implementation of CATs in a more generalized fashion. Consequently, **A-VLAT** and **A-CALVI** assessments

are designed such that they can be reshaped by intermediate programmers, making it easier for researchers and practitioners to adapt these assessments for specific needs. As more assessments are developed in the visualization community, such infrastructure could play a role in developing adaptive versions of future tests, while also providing helpful validity, reliability, and general test analysis capabilities.

3.7.5 Automated Question Generation

Although **A-VLAT** and **A-CALVI** are half the lengths of their original versions, the size of the question banks are still limited, meaning that repeated administrations of the tests would increase the question exposure quickly. Larger banks can reduce question exposure and allow for more attempts of re-assessments without giving out the same questions. However, generating large banks is costly, both in terms of time and resources. The current process is not scalable because it requires manually creating the visualizations and writing the question text for each question, as well as recruiting a large number of participants for test tryout in order to estimate the item parameters needed for adaptive testing.

There is an opportunity for automating the question generation process and inferring item parameters without recruiting a large number of actual participants: the questions in the banks are made up of specific components that are common in all questions, such as the visualizations, question text structure, and the answer options. One can potentially leverage machine learning techniques to automatically generate questions and infer their item parameters using substantial existing data. In Chapter 4, we introduce VILA, a pipeline that uses large language models to generate visualization questions at scale.

Chapter 4

VILA: Scalable Generation of Visualization Questions with LLMs

This work is done in collaboration with Lily Ge, Yiren Ding, Lane Harrison, Fumeng Yang, and Matthew Kay. This chapter is published as a full paper at the IEEE VIS Conference in 2024 [39].

4.1 Introduction

Visualization items (questions)—factual questions about visualizations that ask viewers to accomplish visualization tasks—are regularly used in the field of information visualization as educational and evaluative materials [81, 55, 15, 67, 129, 27]. Teachers of visualization courses require an extensive pool of materials for quizzes, practice problems, and exams; researchers require items to study and advance visualization literacy [81, 55, 38, 67]; and developers of visualization question-answering (VQA) systems seek items as training and test data [70, 91, 96].

Many of these applications of visualization items require large and diverse item banks. For example, visualization literacy researchers may want to track skill development over time or to evaluate intervention efficacy through pre- and post-testing; both

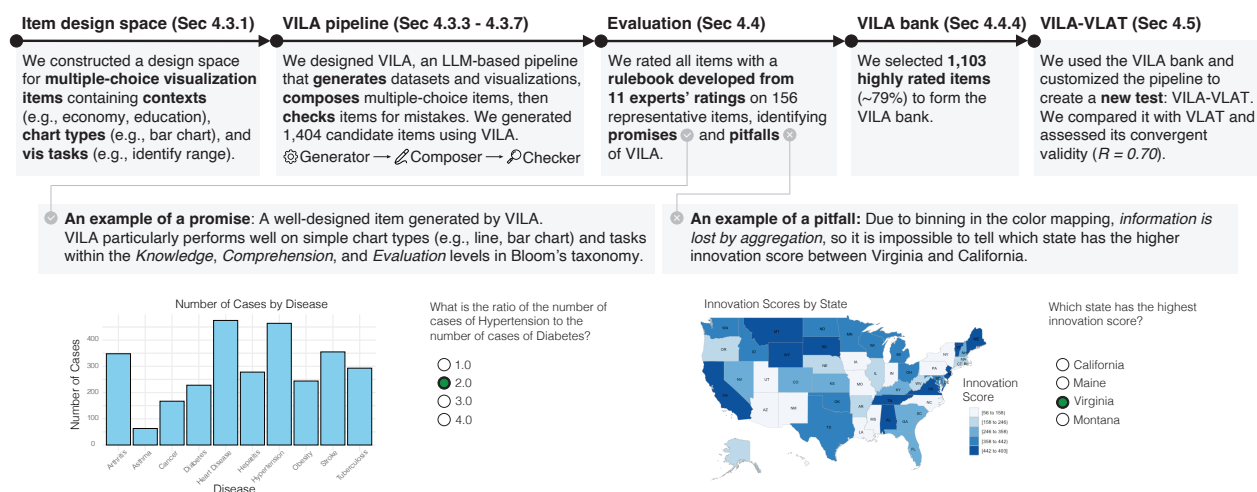


Figure 4.1: Overview of this paper: developing the VILA pipeline, evaluating the candidate bank, and demonstrating a potential application—the new VILA-VLAT visualization literacy test.

require large banks to avoid item overuse (the same item being administered to the same person multiple times). Visualization educators need large item banks to reduce the labor of repeatedly creating new materials and to avoid the risk of leaking homework or test materials. Item banks must also be diverse to ensure that measurement is not biased by the properties of any particular item or specific context, and educators need diverse item banks so that they can assess students' skills across a range of cognitive levels. For example, having items that cover all levels of Bloom's taxonomy [76] (a widely-used taxonomy of learning goals) would help students develop—and teachers assess—a variety of skills, all the way from simple value retrieval tasks to high-level tasks like judging the appropriateness of a visualization design.

However, the generation of high-quality, diverse banks of visualization items is a resource-intensive task. Significant expertise is needed to develop well-designed visualizations, craft questions that accurately assess intended competencies, and curate multiple-choice answers with one best option [55, 81, 15]. There is an untapped need for a way to generate high-quality, diverse items more efficiently and at scale.

Recent advancements in LLMs have demonstrated their capacity to perform tasks including text generation, code generation, and question answering, with promising levels of success [1, 28, 21, 43]. Creating a visualization item requires similar abilities such as generating code for a visualization and writing questions based on it. Thus, LLMs may be able to produce diverse, high-quality visualization items at scale.

To investigate the promises and pitfalls of LLM-based generation of multiple-choice visualization items, we:

1. **Develop the VILA (Visualization Items Generated by Large Language Models) pipeline**, an LLM-based pipeline for generating large, diverse banks of multiple-choice visualization items. Through an iterative design process, we devise a three-stage pipeline: ⚙️ Stage I: Generator generates a data visualization and its associated dataset; ✍️ Stage II: Composer composes the question, options, and correct answer of the multiple-choice item; and 🔍 Stage III: Checker checks the generated item and regenerates a new one if mistakes are detected.
2. **Use the VILA pipeline to construct the VILA bank**, a bank of 1,103 multiple-choice visualization items covering 9 contexts (underlying topics of data), 12 chart types, and 13 visualization tasks covering all levels of Bloom’s taxonomy. We use the VILA pipeline to generate 1,404 candidate items and then evaluate them based on our rulebook with 11 visualization experts’ input, yielding the VILA bank.
3. **Identify promises and pitfalls of the VILA pipeline** based on the evaluation of the VILA bank. We find the VILA pipeline can reliably generate items with the most common chart types (e.g., *line chart*, *bar chart*, and *area chart*) and items with tasks on the *Knowledge*, *Comprehension*, and *Evaluation* levels of Bloom’s taxonomy. We also

identify six classes of errors that the VILA pipeline made, such as *Disregards Perceptual Limits* and *Information Lost by Aggregation*, highlighting the current limitations of this approach.

4. **Use VILA to create a new visualization literacy test, VILA-VLAT**, which measures the ability to read and interpret visualizations using the same tasks and chart types as the existing VLAT (Visualization Literacy Assessment Test) [81]. Results of an online study comparing VILA-VLAT and VLAT show moderate-to-high convergent validity ($R = 0.70$), demonstrating the potential of our approach.

Our work highlights the need to precisely understand the promises and pitfalls of LLMs for visualization item generation and to integrate human oversight into their use. We provide recommendations on how to use the VILA pipeline and discuss application areas that can benefit from our work. Our work just scratches the surface of the potential for LLMs, used appropriately, to help generate diverse, high-quality visualization items at scale, addressing a persistent barrier that many research and education applications currently face.

4.2 Background

4.2.1 Multiple-Choice Visualization Items

Visualization items are used in the field of information visualization to measure skills and abilities. In particular, multiple-choice visualization items are prevalent in the development of visualization literacy assessments [15, 81, 55]. Boy et al. [15] developed a set of tests to assess people’s ability to interpret line charts, bar charts, and scatterplots. Following this work, Lee et al. [81] developed a Visualization Literacy Assessment

Test (VLAT) that contains 53 multiple-choice items across 12 chart types. Ge et al. [55] expanded prior definitions of visualization literacy to incorporate the ability to detect visualization misinformation, creating the Critical Thinking Assessment for Literacy in Visualizations (CALVI), which has a bank of 45 multiple-choice items. Outside of assessment development, Huynh et al. [67] used multiple-choice items in their educational role-playing games to support visualization literacy education in young children.

Multiple-choice visualization items are also used in a wide range of experiments outside of visualization literacy. For example, Tandon et al. [129] used multiple-choice items to study the effects of domain and spatial visualization ability on people’s performance on information visualization tasks; Chen et al. [27] designed multiple-choice items to compare different techniques of visualizing cyclical data; and Lee-Robbins et al. [82] used multiple-choice items in a large-scale experiment to study how specification formats impact visualization design. Moreover, researchers who develop VQA systems use visualization items to train and test computer vision models that answer questions based on the images of data visualizations [91, 70, 96].

4.2.2 Automatic Question Generation in Visualization Research

Automatic question generation (AQG) generally consists of two steps: first, question developers create item models or templates that characterize elements in the questions that are subject to change. Second, different values are inserted into these templates to generate new but similar questions [61]. This approach has also been used by VQA researchers to create datasets for training and testing purposes: Kafle et al. [70], for example, used fixed templates to generate bar chart items. Methani et al. [96] recruited crowdsourced workers to generate questions for given visualizations, then extracted templates from these questions to generate more items, and asked in-house annotators to check and para-

phrase the items. Although a large number of items were generated by this approach, many have answers that only come from a small fixed-sized vocabulary (e.g., “Yes” and “No”) and do not cover visualization tasks that involve various operations such as comparison and aggregation [91].

Recent advances in large language models have drawn visualization researchers to explore their capabilities and applications in AQG. For the purpose of training VQA models, Masry et al. [91] used T5 to create a set of 23,100 free-response visualization items based on human-written chart summaries of existing charts crawled from online sources. The items are question-answer pairs across three types of visualizations (*line chart, bar chart, pie chart*) and 4 types of questions (*data retrieval, visual, compositional, both visual & compositional*); the authors evaluated $\sim 5\%$ of this set manually and found 86.4% to be answerable and correct.

Inspired by this work, we apply LLMs to generating multiple-choice visualization items covering a wider variety of tasks, chart types, and all cognitive levels of Bloom’s taxonomy. We also conduct a thorough evaluation of the types of mistakes LLMs currently make when generating these types of items.

4.2.3 LLMs for Visualization Education

Data visualization researchers have begun to explore the potential of LLMs in visualization education. Chen et al. [29] evaluated GPT’s ability to complete homework and quizzes in a college-level data visualization course. They showed that GPT is capable of completing a range of visualization tasks, such as data interpretation and visualization design, at a promising success rate (GPT-4 scored 80% on the course assignments). Chen et al. also identified limitations of GPT, such as its inability to read visualizations accurately and

lack of creativity.

In another empirical study, Joshi et al. [69] evaluated LLMs’ ability in educating novices about Parallel Coordinate Plots (PCPs) using Bloom’s taxonomy. They showed that while LLMs are capable of generating useful and relevant recommendations to teach people about PCPs, some recommendations are completely inappropriate for PCPs and many do not fall under the proper levels of Bloom’s taxonomy. Therefore, they recommend human expert oversight of LLM output.

4.2.4 Effective Prompt Design for LLMs

A *prompt* refers to the instructions that guide an LLM’s response. As the quality of the prompt greatly affects the quality of the model’s response [118], researchers and LLM developers have created and evaluated various prompting strategies. Wei et al. [142] introduced *chain-of-thought* prompting, which works by decomposing multi-step problems into intermediate steps. They showed this approach improved LLM performance on a range of reasoning tasks. Building on the *chain-of-thought* design, Press et al. [111] developed the *self-ask* strategy, which requires the LLM to state a series of follow-up questions and answer them. LLM developers such as OpenAI have also published recommendations regarding the design of prompts [102]. All of these prompt designs share two common core principles—(1) specify the steps needed to fulfill the user’s request and (2) separate complex tasks into simpler subtasks—which guided us in the design of our prompts.

4.3 Item Generation

Our goal is to create a method capable of generating large, diverse banks of multiple-choice items that span a wide range of chart types, visualization tasks, and contexts. We

first compiled comprehensive lists of these characteristics, which formed our item design space. Next, we used an iterative design process to create a three-stage LLM-based item generation pipeline, the VILA pipeline. We then used it and the item design space to create a candidate bank of 1,404 items.

4.3.1 Design Space of Items

Item Definition. Figure 4.3 is an example multiple-choice visualization item. On the left side is a bar chart on energy production by source, which is the **visualization component** of the item. On the right side of the chart is the **textual component**: the **question stem** asks a viewer to identify the source with the highest bar; the question stem is followed by the **options** and the **correct answer**.

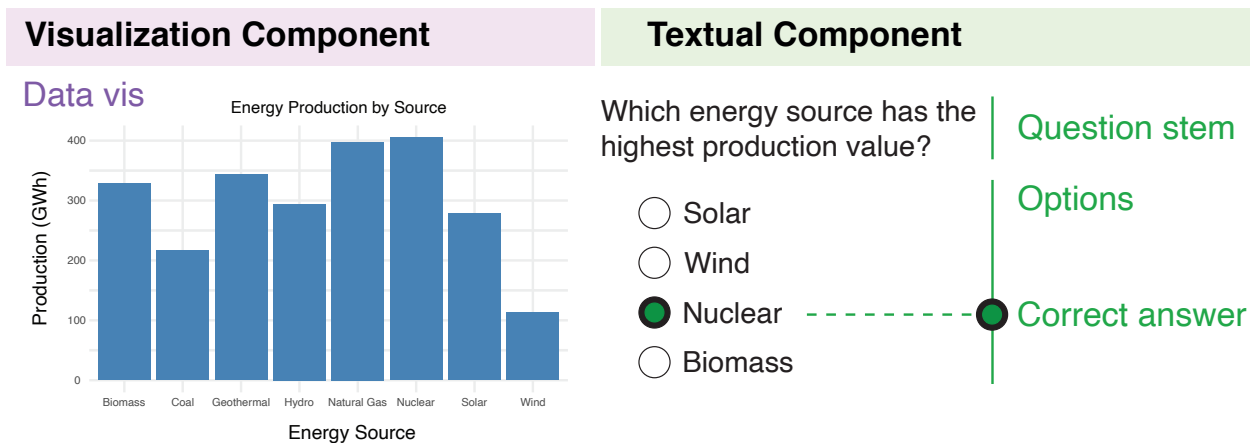


Figure 4.3: An example multiple-choice visualization item with its components labeled.

The visualization component can be described by its **chart type** and **context**, where context is the underlying topic of the data. For example, the chart type in Figure 4.3 is *bar chart* and the context is *energy and environment*. In the textual component of Figure 4.3, the question stem asks the viewer to accomplish the **visualization task**, *locate value*, by choosing the correct answer from the options. The combination of (**context**, **chart type**, **visualization task**) characterizes a multiple-choice visualization item, and

forms the foundation of our item design space. Below, we describe how we created a diverse and representative list for each of these three dimensions.

Context. To compile a list of contexts where data is prevalent in the real world, we adapted the topics from Our World in Data [104], modified the names of the topics for brevity, and created a list of 9 contexts for our design space: *economy, education, energy & environment, food & agriculture, health, innovation, politics, population & demography, and weather & climate*.

Chart Type. We selected the 12 chart types used in VLAT [81]: *area chart, bar chart, bubble chart, choropleth map, histogram, line chart, pie chart, scatterplot, stacked area chart, stacked bar chart, tree map, and 100% stacked bar chart*.

Visualization Task. To compile a diverse range of visualization tasks, we adapted a visualization task taxonomy from Burns et al. [22], where they categorized visualization tasks into Bloom’s taxonomy.

Considering our need to create multiple-choice items, we revised the tasks in this taxonomy using the criterion of *can a human visualization expert create a multiple-choice item of this task with reasonable quality?*. We renamed some tasks to improve clarity (e.g., *calculate the difference between two points* → *estimate the difference between two values of the same type*), added additional common tasks from another taxonomy [3] that are sufficiently different from existing ones (e.g., *identify range*), and removed tasks that are not suitable for the multiple-choice format (e.g., *translate the data in a chart to a table*). We combined *describe a trend* and *describe the relationship between two variables* into *describe trend or correlation* because they are similar; we also expanded *judge which design is more appropriate* into *judge which task this visualization design best supports* and *judge which*

visualization design is more appropriate for a task because they are sufficiently different. The resulting taxonomy below contains 13 visualization tasks across Bloom's taxonomy's 6 cognitive levels:

1. **Knowledge:** *identify labels of scales, identify range, locate value, make comparisons, retrieve value.*
2. **Comprehension:** *describe the topic of the visualization.*
3. **Application:** *estimate average, estimate difference, estimate ratio.*
4. **Analysis:** *describe trend or correlation.*
5. **Synthesis:** *describe the characteristics of an alternative chart type.*
6. **Evaluation:** *judge which task this visualization design best supports, judge which visualization design is more appropriate for a task.*

Details of our revision can be found in the *Revisions of Burns' Taxonomy* in the OSF repository.

While some visualization tasks may not always be appropriate for certain chart types, we examined all combinations of chart type (12) and visualization task (13) and confirmed that it was possible to design a meaningful, quality item for every combination ($12 \times 13 = 156$). Therefore, our design space contains all 9 contexts $\times$ 12 chart types $\times$ 13 visualization tasks = 1,404 possible combinations.

4.3.2 Design Process for the VILA Pipeline

We took an iterative approach to design the VILA pipeline and to engineer the prompts. We made incremental changes to the prompts, used the current pipeline to generate a small number of items, evaluated the results, and iterated until the results could not be significantly improved. None of the test items generated in this phase were included in the final bank. Three important junctures led to significant changes to the pipeline design, and we describe them below.

Separating Visual and Textual Components Generation. We began by using a single-stage solution, prompting the LLM to write code that generates a synthetic dataset, a visualization, and the textual component of an item. The code would use random number generators to create a dataset, and while it also used seeds to ensure reproducibility, the LLM cannot execute the code itself to create the actual dataset before making the textual component of the item. As a result, the LLM-selected correct answer in the item was often wrong due to the absence of the actual dataset. Therefore, we separated item generation into two stages: in ⚙️ Stage I: Generator, we prompt the LLM to generate code that creates a synthetic dataset and a visualization based on it. We execute the code to create the dataset and the data visualization locally. Then, in ✍️ Stage II: Composer, in a new request to the LLM, we provide the dataset and prompt it to compose the textual component of the item.

Switching from Python to R. We began by prompting the LLM to write code for the visualizations using Python and the `matplotlib` and `seaborn` libraries. We observed that, despite our instructions in the prompt, the resulting visualizations often violated basic design principles such as not overcrowding axes with labels. Therefore, we experimented with using R and the `ggplot2` library. We observed a noticeable improvement in aesthetics and

legibility and kept this change.

Adding Automatic Self-Inspection. Prior work [97] shows that LLMs have the potential to diagnose and fix their own mistakes, so we developed a third stage that reuses the LLM to inspect the items. We found preliminary evidence supporting this claim through our iterative design process, so we created 🕒 Stage III: Checker, where we pass the item generated by ⚙️ Stage I and ✍️ Stage II to the LLM, prompting it to check the item and regenerate the item if mistakes are found.

4.3.3 Overview of the VILA Pipeline

Our iterative design process yielded the three-stage VILA pipeline: ⚙️ Stage I: Generator creates a dataset and a visualization based on the given context and chart type; ✍️ Stage II: Composer creates the textual component of the item based on the output from the previous stage and the given visualization task; and 🕒 Stage III: Checker inspects the item and regenerates the textual component if mistakes are detected. The prompt summary for each stage is shown in Figure 4.4.¹ Each run of a stage involves sending the LLM the prompt with specific inputs and obtaining one respective response (outputs); to obtain multiple responses based on different inputs, a stage needs to be run multiple times and runs of a stage are independent. The three stages are also independent in that a stage does not have access to the prompt of the previous stage. This design compartmentalizes the item generation process and enables users to customize it based on their needs. For instance, if an item developer wishes to use or build their own specific datasets and data visualizations, they could modify or skip ⚙️ Stage I and start with ✍️ Stage II. We demonstrate this type of customization in Section 4.5.

¹The full prompts for the three stages are in the OSF repository of this project.

We chose GPT-4 (`gpt-4-1106-preview`), a state-of-the-art LLM for generating code and answering exam questions [1]. This GPT-4 version also supports a seed parameter that improves the reproducibility of results [103]. We set the temperature parameter to be 0.2 in all three stages to achieve consistent and coherent results [101]. Details of our model specifications are in the OSF repository.

4.3.4 Stage I: Dataset and Visual Component Generator

In this stage, we provide a `context` and a `chart type` of the visualization as input to the LLM. In each run, we prompt it to generate the R code for creating a synthetic `dataset` in the given context and the `vis code` for rendering the given chart type from the dataset. Figure 4.4.A contains the prompt summary of Stage I. The prompt has two parts: (1) a generic part applied every time and (2) a chart-type specific part for each chart type (line A19). The content of our prompt contains requirements on the output (dataset and visualization), which can be summarized into the following categories: consistency (lines A1–A2), compatibility (lines A9 & A11), completeness (lines A14 & A17), legibility (lines A15–A16), and diversity (line A19).

For more complex chart types such as *choropleth map* and *tree map*, prompting the LLM to use specific libraries (`choroplethr` and `treemapify`) yielded code with satisfying outputs. Also, certain chart types require prompting the LLM to use specific `ggplot2` commands to ensure correctness: for example, *100 % stacked bar chart* requires prompting the LLM to include a specific line of code to ensure that the labels on the y-axis range from 0% to 100%. In addition, when *month* (e.g., *Jan*) was on the x-axis, it was often presented alphabetically despite our explicit instruction requiring temporal values to be displayed chronologically; therefore, we prompted the LLM to avoid using *month* in the dataset.

(A) Stage I: Generator prompt	
1 You generate datasets and make data visualizations based on the datasets and the user's specific requirements. 2 You use R programming language and, unless specified, only use libraries including "ggplot2", "tidyverse", and "dplyr" to generate the datasets and data visualizations. 3 ▶ # instructions for output formats omitted for space, see supplements 4 I will give you a context and a chart type of data visualization. 5 I want you to use R programming language to make a synthetic dataset in the given context and a data visualization of the specified chart type based on the synthetic dataset. 6 Here is the context: <code>context</code> 7 Here is the type of data visualization: <code>chart type</code> 8 The values in the synthetic datasets must have meaningful names in the given context. 9 All values in the dataset must be whole numbers. 10 Make sure to generate a dataset that is well suited for the chart type. 11 Next, I will give you additional requirements for the visualization and the dataset. 12 The data visualization must be well-designed with the following requirements: 13 (1) There should be a title for the visualization. Display the title at the top. 14 Do not overlay the title on other components of the visualization. 15 (2) Do not overcrowd color legends or axes with labels. 16 (3) Leave enough margins around the visualization so that all labels are displayed fully. 17 (4) All data points in the dataset must be displayed on the visualization. 18 (5) Do not use month as a type of value in the dataset. 19 For <code>chart type</code>, <code>chart specific requirements</code> 20 ▶ # instructions for output formats omitted for space, see supplements	Consistency Visualization is stylistically consistent Inputs Compatibility The data suits the chart type Compatibility Completeness Show all info about the dataset Legibility Visualization is clear Completeness Diversity Variety of patterns within chart types
(B) Stage II: Composer prompt	
1 I will first give you the R code as a string. 2 Here is the R code: <code>vis code</code> 3 I will now give you the dataset the R code generated as a string. 4 Here is the dataset: <code>dataset</code> 5 Write a multiple-choice item associated with the data visualization and the dataset. 6 The multiple-choice item must have the following properties: 7 (1) the task the viewer does is to the task the viewer does is to <code>vis task</code> ; <code>vis task description</code> 8 (2) the item has only one correct answer. 9 (3) the viewer should be able to find the correct answer with only the visualization, question stem, options, and without access to the dataset. 10 First, generate a question stem that allows a tester taker to do the task of "<code>vis task</code>" with the data visualization created by the R code. 11 Second, generate one correct answer to the question stem. 12 Use the provided dataset to find the correct answer. 13 Third, generate three other unambiguously incorrect options for the question stem and the data visualization. 14 Do not include justifications for the options in the options themselves. 15 ▶ # instructions for output formats omitted for space, see supplements 16 Make sure the correct answer is one of the options in part 2 of your response. 17 In the question stem of part 1: 18 (1) do not include the phrase "R code" or "R script"; the question stem should not reveal how the dataset or visualization is generated. 19 (2) the question stem should be succinct; do not use the phrases "According to the visualization", "Based on the visualization", or similar phrases. 20 ▶ # instructions for output formats omitted for space, see supplements	Inputs Explanation of Task Instructions creating question stem and options
(C) Stage III: Checker prompt	
1 I will give you a multiple-choice item on a data visualization task which consists of six parts: 2 part 1: the question stem 3 part 2: options 4 part 3: the correct answer 5 part 4: the R code that generated the dataset and the data visualization 6 part 5: the realized dataset from the R code (no more randomness) 7 part 6: the visualization task 8 Here are the first three parts: <code>question stem</code> <code>options</code> <code>correct answer</code> 9 Here is the visualization task (part 4): <code>vis task</code> ; <code>vis task description</code> 10 Here is the R code that generated the dataset and the data visualization: <code>vis code</code> 11 Here is the dataset: <code>dataset</code> 12 I want you to check whether there are any mistakes in this item. Specifically, 13 (1) check that the correct answer is indeed correct based on the question stem, the visualization task, and the dataset. You must find the correct answer yourself first and then compare it to the current correct answer. 14 (2) check that the correct answer is the only one that is correct among all the options. The incorrect options must be clear and unambiguously incorrect. 15 ▶ # instructions for output formats omitted for space, see supplements 16 If you detect a mistake in an item, use the dataset, the R code for the visualization, and the visualization task to make a new item so that there are no more mistakes. 17 ▶ # instructions for output formats omitted for space, see supplements	Inputs Check correctness Check uniqueness Check and remake

Figure 4.4: The prompt summary from the three stages of the VILA pipeline. The variables are replaced by a different value each time the prompt is used. The inputs, requirements, and instructions at each stage are annotated on the right.

4.3.5 Stage II: Textual Component Composer

In this stage, we provide three inputs to the LLM: (1) the **dataset** from ⚙️Stage I, (2) the **vis code** from ⚙️Stage I, and (3) a **vis task** from our list of 13 tasks in the revised taxonomy. In each run, we prompt the LLM to output a **question stem**, its **options**, and the **correct answer** for a single item (see Figure 4.4.B).

Similar to ⚙️Stage I, the prompt in ✍️Stage II consists of a generic component applied to all items and a task-specific component (line B6). The content of the prompt contains the explanation of the visualization task (line B6) and instructions for how to generate the correct and incorrect options (lines B7–B20).

The prompting strategy of decomposing a multi-step problem into sub-problems [142, 102] improved the performance of this stage, especially for value-related tasks including *retrieve value*, *identify range*, and *estimate difference/ratio/average*, where creating the textual component can be divided into a series of specific arithmetic operations on how to find the correct answer and how to create the incorrect options.

4.3.6 Stage III: Item Checker and Regenerator

In this stage, we provide (1) the **dataset** and **vis code** from ⚙️Stage I, (2) the **vis task**, and the (3) the textual component (**question stem**, **options**, **correct answer**) from ✍️Stage II. In each run, we prompt the LLM to output a new textual component (**question stem**, **options**, **correct answer**) for a single item if a mistake is detected and otherwise the same textual component from the input.

Figure 4.4.C contains the prompt summary of 🔍Stage III: Checker. The prompt again

consists of a generic part that is applied to all items and a task-specific part (line C9). The content of the prompt contains two inspection instructions that prompt the LLM to check whether the correct answer is indeed correct (line C13) and whether it is the only correct answer (line C14). It also contains a request to regenerate an item when a mistake is detected (line C20). We present an analysis of the LLM’s self-inspection performance in Section 4.4.6.

4.3.7 Creating the Candidate Bank of Items

We used the VILA pipeline to create a candidate bank of items. We ran ⚙️ Stage I 108 times ($9 \text{ contexts} \times 12 \text{ chart types}$) to create 108 data visualizations ([vis code](#) and [datasets](#)). For each of these 108 outputs, we ran ✍️ Stage II 13 separate times to create the textual components for 1,404 items. These items were checked by 🔍 Stage III, yielding 1,404 LLM-inspected candidate items that span all combinations of context, chart type, and visualization task. Figure 4.5 shows 6 example items in the candidate bank across 6 cognitive levels of Bloom’s taxonomy.

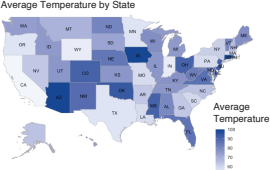
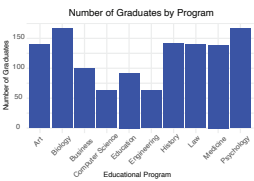
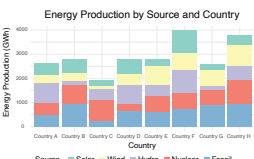
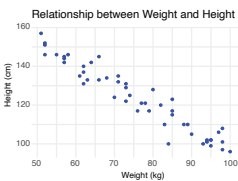
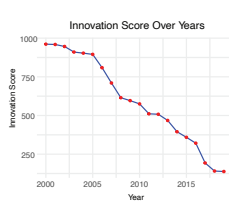
Knowledge	Task: Make comparisons	Context: Weather and climate
Average Temperature by State 	Which state has a higher average temperature, Florida or New York? ☒ Florida ☐ New York ☐ Both have the same average temperature ☐ Cannot be determined from the map	
Number of Graduates by Program 	What is the subject of the bar chart? ☐ The average salary by educational program ☒ The number of graduates by educational program ☐ The number of enrolled students by educational program ☐ The number of faculty members by educational program	Comprehension
Energy Production by Source and Country 	What is the ratio of energy production from Solar to Wind for Country A? ☐ 0.6 ☒ 1.6 ☐ 2.6 ☐ 3.6	Application
Relationship between Weight and Height 	What is the trend or correlation between Weight and Height as depicted in the scatterplot? ☐ A strong positive correlation between Weight and Height. ☒ A strong negative correlation between Weight and Height. ☐ No correlation between Weight and Height. ☐ A moderate positive correlation between Weight and Height.	Analysis
Innovation Score Over Years 	If the trend shown in the line chart were to be displayed using a bar chart, which of the following characteristics would accurately describe the bar chart? ☐ Each bar would represent a year, with bar height increasing from 2000 to 2019. ☐ Each bar would represent a year, with bar height remaining constant across all years. ☒ Each bar would represent a year, with bar height decreasing from 2000 to 2019. ☐ Each bar would represent a year, with colors of the bars representing different innovation	Synthesis
Distribution of Food and Agriculture Sectors 	What task does this pie chart design best support? ☐ Comparing the exact numerical values between different food and agriculture sectors ☒ Showing the distribution of various food and agriculture sectors as a part of a whole ☐ Displaying the trend of agricultural sector growth over time ☐ Identifying the correlation between different agricultural sectors	Evaluation

Figure 4.5: Example items from the candidate bank across 6 levels of Bloom's taxonomy. The task names are paraphrased for brevity. The highlighted option is the LLM-selected correct answer.

4.4 Item Evaluation

We examined the quality of all candidate items. First, we developed a set of evaluation questions to guide the process. We then conducted a two-phase evaluation with these questions. In Phase I, we created a rulebook for answering the evaluation questions based on 11 visualization experts' input. In Phase II, two author-evaluators followed the rulebook and the questions to evaluate the entire candidate bank. Our evaluation resulted in a final bank of 1,103 items and a set of limitations of the LLM for generating multiple-choice visualization items.

4.4.1 Evaluation Questions

Four authors discussed the most important aspects of the quality of our items. We iterated on the question wording and arrived at four:

1. **Relevance:** *How relevant is this item to assess people's ability to carry out the visualization task?* This is rated on a scale from 1 (*Not relevant*) to 4 (*Highly relevant*).
2. **Clarity:** *How clear are the question statement and the options?* This is also rated on a scale from 1 (*Not clear*) to 4 (*Highly clear*).
3. **Answerable:** *Is one of the options a correct or approximately correct answer to the question?* This is rated with *Yes* and *No* options.
4. **Correctness:** *Is the answer marked as correct indeed the correct answer?* This is rated with three options: *Yes*, *No*, and *More than one option could be considered correct*.

Relevance is connected to content validity [110], which evaluates how well an item mea-

sures what it aims to measure; *Clarity* ensures that readers can understand the item clearly; *Answerable* and *Correctness* are necessary measures that ensure the correct answer can be identified by a reader and it is the best option among available options.

4.4.2 Phase I: Development of the Rulebook

The goal of this phase is to create a rulebook that will allow two author-evaluators to evaluate the entire candidate bank using criteria grounded in a larger set of experts' opinions. We invited external data visualization experts to evaluate a representative sample of items from the candidate bank, together with the two author-evaluators.

Experts. We used a convenience sample of 11 expert participants. They came from six universities and four industry organizations; none were at the same institution(s) as any author. All have obtained a Ph.D. in visualization-related fields.

Representative Sample of the Bank. Due to the large size of the bank, we selected a representative sample of 156 items, as a balance between comprehensiveness and practical time constraint, to help create the rulebook. We randomly selected one item from each of the $12 \times 13 = 156$ (chart type, task) combinations, while also ensuring that each of the 9 contexts is represented approximately equally.² Each expert received 52 of the 156 items, which included all 13 types of tasks with 4 items for each task. Experts were asked to rate these items based on the 4 evaluation questions and write optional comments. Each item was evaluated by at least 3 experts. The two author-evaluators also evaluated all 156 items independently.

To analyze the evaluations of the experts and author-evaluators for the purpose of creating a rulebook, we focused on cases with disagreements and low ratings. For in-

²The representative sample of items and the code used to select them are provided in the OSF repository.

stance, we examined all items that at least one expert did not rate *Yes* for *Answerable* or *Correctness*. For such items, we looked at the ratio of *Yes* to non-*Yes* ratings, read the experts' comments (if any), and decided on a final rating. We also documented the justifications for our final ratings and how to handle similar situations as part of the rulebook.

Rulebook Construction.³ The rulebook contains both rating suggestions and guidelines for each (chart type, task) combination.

For the *Relevance* and *Clarity* ratings of a (chart type, task) combination, we computed the average ratings of these two categories, respectively, across the experts and the two author-evaluators. We took the floor and the ceiling of the average ratings and used them as the recommended rating options for items with this (chart type, task) combination. The rulebook contains suggested ratings for these two evaluation questions for all (chart type, task) combinations.

The *Answerable* and *Correctness* categories require more scrutiny depending on the specific item, so we do not provide specific suggested ratings for all items with a (chart type, task) combination. Instead, we identify combinations that are prone to certain types of errors to help author-evaluators make decisions. For example, the combination (*choropleth map*, *estimate difference*) tends to have multiple correct answers, because precise values are lost through aggregation on the color scale; the rulebook reminds the author-evaluators to be mindful of such situations and give ratings on a case-by-case basis.

4.4.3 Phase II: Evaluation of the Candidate Items

In this phase, the two author-evaluators each evaluated half of the candidate items. When they encountered situations not documented in the rulebook, they discussed to reach an

³The complete rulebook is available in the OSF repository.

agreement and added it to the rulebook. They also documented cases with ratings against the rulebook’s recommendations and their justifications for those decisions.

4.4.4 Final Bank (The VILA Bank)

We used the following **inclusion criteria** to select the final bank of items: items must have a rating of 3 or 4 for both *Relevance* and *Clarity* and a rating of *Yes* for both *Answerable* and *Correctness*.

Figure 4.6 summarizes the VILA bank. Each cell indicates the number of items in that (chart type, task) combination in the VILA bank. The VILA bank contains a total of 1,103 items (out of 1404) across 154 (chart type, task) combinations, which is $\sim 79\%$ of the candidate bank. All items have *Yes* ratings to both *Answerable* and *Correctness*. The average *Relevance* and *Clarity* ratings for the bank are 3.7 and 3.8, respectively. The VILA bank is available in the OSF repository.

4.4.5 Promises and Pitfalls

Through this evaluation, we also identified the strengths of the VILA pipeline and classified the errors that it is prone to make.

4.4.5.1 “Promises” of the VILA Pipeline

From Figure 4.6, we observe that the VILA pipeline is capable of reliably producing high-quality items with many chart types and tasks. In particular, it performs well on generating items with the most common and simplest chart types, such as *line chart*, *bar chart*, *pie chart*, and *area chart*. It also performs well on creating items with tasks in the *Knowledge* and *Comprehension* cognitive levels. Moreover, we note that the VILA pipeline can reliably

		Area	Bar	Bubble	Map	Histogram	Line	Pie	Scatterplot	Stacked Area	Stacked Bar	Treemap	100% Bar
													
Knowledge	Make comparisons	9	9	6	7	8	8	9	6	6	4	5	5
	Retrieve value	9	9	7	7	1	9	9	6	7	9	6	9
	Locate value	9	9	9	5	6	8	9	9	5	8	5	8
	Identify range	7	9	7	8	8	9	8	9	3	6	1	8
	Identify labels	9	9	9	8	9	8	4	9	7	8	6	9
Comprehension	Describe topic	8	9	9	9	9	9	9	9	8	9	6	9
Application	Estimate difference	8	8	3	0	4	9	8	6	3	5	4	8
	Estimate ratio	8	9	4	4	3	8	7	2	8	8	5	9
	Estimate average	9	7	6	0	7	7	9	7	2	6	5	9
Analysis	Describe trend	9	2	6	8	7	9	1	8	7	8	1	8
Synthesis	Describe alter. chart	9	9	1	8	3	7	9	4	6	3	6	7
Evaluation	Choose vis for task	9	9	9	9	9	9	8	9	8	8	6	8
	Choose task for vis	8	9	9	8	9	8	9	9	8	8	6	9

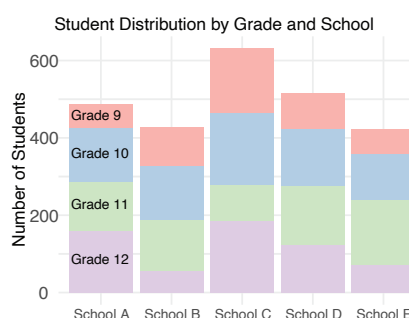
Figure 4.6: Summary of the final bank of 1,103 items. Names of chart types and visualization tasks are paraphrased for brevity. Each cell represents a (chart type, task) combination and shows the number of items with that combination in the final bank. The maximum is 9 because there are 9 contexts in our item design space.

generate items with tasks on the highest cognitive level in Bloom's taxonomy, *Evaluation*, which include the tasks *judge which visualization design is more appropriate for a task (choose vis for task)* and *judge which task this visualization design best supports (choose task for vis)*.

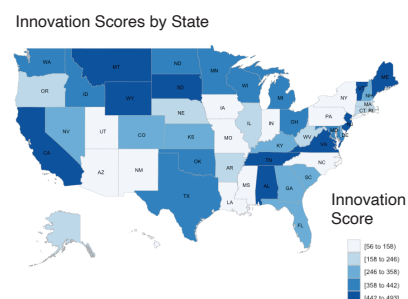
4.4.5.2 “Pitfalls” of the VILA Pipeline

To better understand the limitations of the VILA pipeline, guide others in its use, and help researchers improve upon this method, we classified the errors the pipeline produced into six major categories.⁴

I. Disregards Perceptual Limits. The item disregards humans' perceptual limits and is too difficult to answer. For instance, in the stacked bar chart on the right, it is very hard to visually estimate the average number of students in Grade 9 across the 5 schools because the red segments are not aligned. If the incorrect options are too close to the correct one, this item can be impossible to answer.

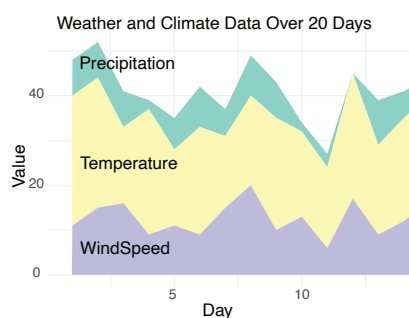


II. Information Lost by Aggregation. Data aggregation in the visualization obscures the correct answer. For instance, identifying the state with the highest innovation score (*locate value*) on the choropleth map on the right is impossible because there are multiple states with the same darkest color due to binning in the color mapping.

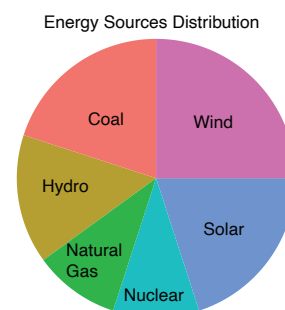


⁴The classification of all errors is available in the OSF repository.

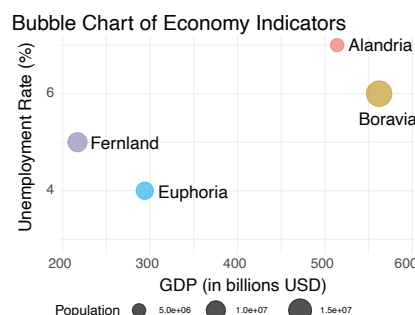
III. Inappropriate Visual-Data Correspondence. The visualizations do not correctly represent relationships in the data, violating the principle of visual-data correspondence [75]. For instance, the stacked area chart on the right stacks wind speed, temperature, and precipitation values together, but the sum of these values on the y-axis is not meaningful.



IV. Task Not Suitable for Chart. The design of the visualization does not support the item's visualization task. For instance, the pie chart on the right does not contain temporal or geographic variables, so the task *describe trend or correlation* is not suitable for this chart.



V. Unclear Phrasing of Question. The question stem is not phrased clearly enough for the reader to identify the data needed to answer the question. For instance, when asked “which country has a larger population size, the one with higher GDP or the one with lower unemployment rate?” for the bubble chart on the right, it is unclear which two countries the question is describing, so the item is not answerable.



VI. Incorrect Answer. The options do not contain a correct answer, or the actual correct answer is not the LLM-selected correct answer. For instance, the histogram on the right has a symmetric distribution, but the LLM-selected correct answer to the question “what pattern does the histogram show?” is “right-skewed”. This may be related to the concept of **hallucination** in language models [68, 71], where the LLM-generated content is nonsensical or unfaithful to the provided source content [68].



While some errors may be fixable by better prompt engineering or different pipeline designs, these classes of errors reveal the limitations of our LLM-based approach. Human experts possess nuanced, often implicit knowledge of how to design good visualizations and good items that the LLM does not currently have. It is therefore crucial to elicit experts’ implicit knowledge and incorporate them into the design process. To this end, we discuss the importance of human oversight in Section 4.6.2 and the potential of human-LLM collaboration in Section 4.6.5.

4.4.6 Performance of Stage III

To investigate the performance of $\textcircled{P}$ Stage III: Checker, we also evaluated the versions of the 1,404 items **before** they were inspected by $\textcircled{P}$ Stage III: Checker. Combined with the evaluation of the candidate bank from $\textcircled{P}$ Stage III, this allows us to examine $\textcircled{P}$ Stage III: Checker’s capability of detecting and fixing mistakes. We defined a mistake as an item that would not pass the inclusion criteria in Section 4.4.4.

Figure 4.7 summarizes the performance of $\textcircled{P}$ Stage III. The accuracy of the pipeline with $\textcircled{P}$ Stage III: Checker is $\frac{958+89+56}{1404} = 0.79$. If we were to use the pipeline without $\textcircled{P}$ Stage III: Checker, the accuracy would be $\frac{958+100}{1404} = 0.75$; if we were to use $\textcircled{P}$ Stage III: Checker and just remove all items it flags as mistakes without fixing them, the accuracy would be $\frac{958}{958+220} = 0.81$. This demonstrates that $\textcircled{P}$ Stage III is capable of improving the VILA pipeline’s performance by fixing items that it deemed to have mistakes, but discarding all such items without fixing is also a reasonable approach with high accuracy.

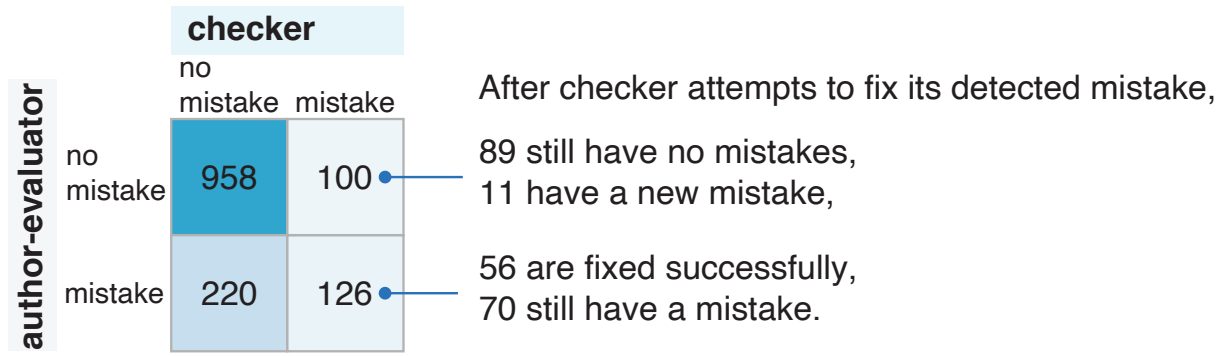


Figure 4.7: Confusion matrix of the performance of $\textcircled{P}$ Stage III: Checker.

4.5 Application Demonstration: VILA-VLAT

In this section, we provide a demonstration of applying the VILA pipeline and the VILA bank to measure visualization literacy. Specifically, by using the final bank and customizing the VILA pipeline, we created and validated a visualization literacy test, VILA-VLAT, that covers the same (chart type, task) combinations as VLAT [81].

4.5.1 Construction of VILA-VLAT

To ensure VILA-VLAT is as comprehensive as VLAT, we used VLAT’s structure to create VILA-VLAT. VLAT contains 53 items that span 12 chart types with 8 tasks [81]. Each VLAT item is a unique (chart type, task) combination, and each test taker is expected to take all

53 items.

Reclassification of VLAT Tasks. The VILA bank has the same 12 chart types as VLAT, but the tasks are not identical. Therefore, we reclassified the task in each VLAT item using our revised task taxonomy. We did so by examining the question stem and the options of each VLAT item to see what task the viewer is asked to accomplish and then looking through our revised task taxonomy to find the most fitting task. This reclassification yielded 46 unique (chart type, task) combinations under our task taxonomy.

Pipeline Customization for Special Items. There are 5 VLAT items whose tasks did not fit in our taxonomy: *identify anomalies* for scatterplot and bubble chart, *identify clusters* for scatterplot and bubble chart, and *identify the hierarchical structure* for tree map. All three tasks are specific to certain chart types, and some require special considerations on the data visualization (e.g., the task *identify anomalies* is only suitable for a scatterplot that contains an outlier).

The VILA pipeline is designed to support using customized visualizations and datasets by modifying ⚙️ Stage I. ✍️ Stage II can also be modified to include additional visualization tasks. To construct these 5 items for VILA-VLAT, we customized the VILA pipeline: for *identify anomalies* and *identify clusters*, we added four chart types (*scatterplot with outlier*, *scatterplot with cluster*, *bubble chart with outlier*, and *bubble chart with cluster*) in ⚙️ Stage I to create $4 \text{ chart types} \times 9 \text{ contexts} = 36$ visualizations. We checked these visualizations and removed 14 that do not contain a visible outlier or cluster. Next, we added the two tasks and their descriptions in ✍️ Stage II's prompt, and the 22 remaining visualizations were then used by ✍️ Stage II and 🔍 Stage III to create and check the items. Then, we evaluated all 22 items using the 4 evaluation questions and decided to keep all, because they passed the inclusion criteria in Section 4.4.4. For *identify the hierarchical structure*, we used the 6

treemap visualizations and datasets from our final bank. We added the task and its description to the prompt in $\mathcal{L}$ Stage II and used $\mathcal{L}$ Stage II and $\mathcal{P}$ Stage III to create the textual components. All 6 items passed the inclusion criteria.

VILA-VLAT contains 51 items from 51 (chart type, task) unique combinations. For each test taker, one item from each (chart type, task) combination is randomly drawn from our final bank and the 28 customized items. VILA-VLAT uses a total of 371 items: 343 items from the VILA bank and 28 customized items.

4.5.2 Validity of VILA-VLAT

The inclusion criteria in Section 4.4.4 ensure that all items in VILA-VLAT have a *Relevance* rating of 3 or 4, which is evidence for high **content validity** [35, 110]. We also evaluate the **convergent validity** [33] of VILA-VLAT by computing the correlation between test-takers' performance on VILA-VLAT and VLAT. Convergent validity is a measure that establishes the equivalence of two tests through strong correlation.⁵

Participants. We recruited 120 participants via Prolific.⁶ We selected our sample size based on similar experiments on convergent validity from Cui et al. [38], anticipating that about 100 of the 120 participants would complete the entire study. All participants fall under the following criteria: speak fluent English, have normal or corrected-to-normal vision, do not have colorblindness, are between 18 and 65 years old, and are U.S. residents. In addition, Prolific participants from similar studies that the authors conducted in the past were excluded.

Procedure. Each participant was asked to take both VLAT and VILA-VLAT,⁷ with a mini-

⁵Our analysis code is preregistered and can be found in the OSF repository.

⁶This study received institutional IRB approval.

⁷Both are implemented using reVISit [44], an open-source software toolkit for creating empirical visualiza-

mum 5-hour break between sessions to avoid fatigue. We counterbalanced to avoid ordering effects: half of the participants (Group I, 60 people) took VLAT first, and the other half (Group II, 60 people) took VILA-VLAT first. In each session, we presented participants with the consent form and information page describing the study structure. We informed them that they needed to select an answer for the current item before proceeding to the next one, and once they moved on, they could not return to previous ones. Participants who failed two or more attention check questions (out of three) were excluded. In the second session, 44 and 45 returned in Groups I and II respectively, for a total of 89 participants. Two participants failed the attention check, leaving 87 participants who completed both tests, and we used their data to compute the correlation for convergent validity.

Results. The correlation coefficient between VLAT and VILA-VLAT was 0.70 with 95% CI of [0.57, 0.79]. Although there is no gold standard, a coefficient below 0.50 should be avoided and above 0.70 is recommended [25], so VILA-VLAT has **moderate to high convergent validity**. In addition, the average correctness rates for VLAT and VILA-VLAT are 0.76 and 0.77, respectively. Together with the convergent validity evidence, this shows that VILA-VLAT is a valid visualization literacy test that measures similar skills as VLAT and has similar difficulty.⁸

tion studies.

⁸Due to an implementation error, two items not in the final bank were administered as part of VILA-VLAT in this study along with the other 371 items. We removed those two items from the responses of participants who encountered them. Details are in the analysis code file of the OSF repository.

4.6 Discussion

4.6.1 Hypothetical Cost Comparison

Is using the VILA pipeline actually worth it? It is hard to provide a universal answer because it depends on the specific needs of different users. Based on the authors' experience on creating visualization items, we present a rough, hypothetical cost comparison for generating the VILA bank to help prospective users reason about the cost of our method and offer people a template to evaluate their specific situations.

Financial Cost. The financial cost of creating items consists of two variables: the number of items needed and the unit cost of an item. The cost of creating the VILA bank of 1,103 items using our pipeline and OpenAI's GPT-4 model is $\sim \$50$, which means the unit cost of an item is $\sim \$0.05$. Two author-evaluators, who are Ph.D. students, evaluated all items in the candidate bank, and it took roughly 8 hours per person (a total of 16 hours). A typical hourly rate of a Ph.D. student in the U.S. is $\$20$; it would cost $\$0.23$ to evaluate one item. Thus, the total cost per item is $\$0.05 + \$0.23 = \$0.28$.

An alternative for creating quality items, for instance, is to hire a Ph.D. student in data visualization to make them from scratch. Suppose they can create 4 context-specific visualizations in an hour; it will take them 27 hours to create 108 visualizations ($9 \text{ contexts} \times 12 \text{ chart types}$). Suppose they can write 10 textual components (question, incorrect options, correct option) of a visualization in an hour; it would take them 140 hours to create 1,404 textual components. The unit price of an item in this approach is $\frac{(27+140) \times \$20}{1,404} = \$2.38$. While there are other alternatives and the specific costs of humans can vary, the VILA pipeline could be a better financial option than manually creating ev-

everything from scratch.

Time Cost. Without parallelization, it took ~ 40 seconds for the VILA pipeline to create an item, so the total time of creating the candidate bank is 16 hours. Together with the 16 hours of evaluation, the total time of creating the VILA bank of 1,103 items is 32 hours, which translates to 34 items per hour. Following our previous assumption, a Ph.D. student can create $\frac{1,404}{27+140} = 8.4$ items per hour, so our method could be advantageous in time compared to an entirely manual approach.

4.6.2 Recommendations for Using the VILA Pipeline

We recommend prospective users of the VILA pipeline first enumerate what chart types, visualization tasks, and contexts their desired items should cover. Users can consult Figure 4.6 and the analysis in Section 4.4.5 to assess whether our pipeline can reliably generate the specific (chart type, task) combinations they require. We advise against using our pipeline to generate combinations that are prone to error. If our item design space does not contain certain chart types, tasks, or contexts that users need, we recommend incorporating these new elements by customizing  Stage I and  Stage II as we did for constructing VILA-VLAT in Section 4.5.1. For (chart type, task) combinations that the VILA pipeline can reliably generate (the dark cells numbered 8 or 9 in Figure 4.6), human oversight may not be necessary depending on the users' requirements and constraints. For (chart type, task) combinations that are less reliable (cells numbered 5, 6, or 7 in Figure 4.6), we recommend users manually evaluate them using the four evaluation questions in Section 4.4.1 and look out for the errors in Section 4.4.5.2. It is also important to consider whether use of VILA with any necessary human supervision will have a cost or time advantage for a particular application (as in Section 4.6.1).

4.6.3 Potential Application Areas

Visualization Literacy. Visualization literacy researchers need large banks of items to reduce item overuse in repeated studies. They also need diverse items to assess people's ability to complete a wide range of tasks across different cognitive levels of Bloom's taxonomy. Adaptive testing of visualization literacy [38], where the next item for a test taker is adaptively chosen based on a person's performance on previous items, can measure people's abilities more accurately and efficiently, but also typically requires large item banks. The VILA pipeline could be used to generate large, diverse banks of visualization literacy items needed for these applications.

Visualization Courses. Teachers of data visualization and statistics classes need evaluative and practice materials to assess students' performance and help them improve. The VILA pipeline offers them an efficient way to generate a wide range of items at scale to reduce manual labor. Having a large item bank also reduces the risk of leaking test materials due to recycling content. The diversity of items the VILA pipeline can generate can also help teachers customize interventions for students with different needs.

Visualization Question-Answering (VQA). VQA researchers require large amounts of question-answer pairs and visualization images to develop their models. The VILA pipeline offers an alternative method of dataset creation compared to using fixed templates or recruiting crowdsourcing participants. In addition, current VQA datasets mainly consist of items with tasks on the lower end of Bloom's taxonomy. As research in VQA advances, the ability of the VILA pipeline to generate items with tasks across a wide range of cognitive levels will help researchers assess the ability of their systems and improve their models.

4.6.4 Design Guidelines for Visualization Items

Throughout the refinement of the VILA pipeline and the evaluation of items, we regularly found the need to explicitly write down otherwise-implicit rules about good item construction so that the LLM was able to apply them. When human developers design visualization items, we often rely on our implicit expertise or have a “hunch” of what makes an item good. For example, if a human expert makes an item that asks the reader to estimate the average value of a line chart, they are aware that the incorrect options cannot be too close to the correct one (see *Disregards Perceptual Limits* in Section 4.4.5.2). However, there is no existing set of explicit guidelines on how to effectively create multiple-choice visualization items. This creates a barrier for people who need to make items but do not have sufficient expertise, and it also makes the item-creation process less systematic. We collected the criteria and instructions from our prompts, as well as the classes of errors from our evaluation, and organized them into design guidelines for creating multiple-choice visualization items. These guidelines are available in our OSF repository.

4.6.5 Design Lessons and Human-LLM Collaboration

It is difficult and time-consuming for human developers to design diverse, high-quality visualization questions at scale. We demonstrate it is possible to partially automate the question generation process and create large sets of diverse questions. While our prompt designs and resulting questions are still imperfect, our work reveals that question generation efficiency can be improved while maintaining quality through human evaluation of questions (which is less costly than pure human question generation; see Section 4.6.1), and that our pipeline can be customized to handle special cases where the LLM otherwise cannot generate questions (see Section 4.5.1).

However, we have only scratched the surface of the potential for question generation using LLMs. We only used one LLM, and it is worth investigating the pipeline’s performance on multiple LLMs or on LLMs fine-tuned for this specific application. We could also incorporate large multimodal models (LMMs) with vision capabilities into the pipeline to generate items based on real-world visualizations. These models may also help address errors we identified in our pipeline, such as *Information Lost by Aggregation* and *Disregards Perceptual Limits*. Also, ⚙️ Stage I and ✍️ Stage II currently are separated, making the visualization generation stage unaware of what types of visualization tasks it needs to support, which can lead to errors such as *Task Not Suitable for Chart*.

In addition to improving the model, we could adopt a more collaborative approach to human–LLM teaming. Currently, human evaluation occurs only after the questions have been generated. A more collaborative, interactive system might seek human feedback continuously throughout the entire process, showing possible visualizations or datasets that experts could select from or refine, feeding back into further visualization or question generation. Or, experts might design question templates interactively, triggering real-time generation of questions that they can quickly see, evaluate, and refine. We believe methods and insights from creativity support tools in HCI [52] and human-AI collaboration [139] could lead to more efficient and reliable question-generation systems.

In Chapter 5, we extend these ideas beyond visualization questions to general educational assessments, developing a human–LLM collaborative system that empowers teachers of different subjects to create, adapt, and refine assessment materials.

Chapter 5

Ripplet: Authoring Educational Assessments through Multilevel Reusable Interactions with LLMs

This work is done in collaboration with Annabel Goldman, Jovy Zhou, Xiaolin Liu, Clarissa Shieh, Joshua Yao, Mia Cheng, Matthew Kay, and Fumeng Yang. This chapter is currently under review at a computer science conference (Nov 2025).

5.1 Introduction

Classroom assessments are critical in education [20, 117, 8]. Formative assessments, such as daily homework and quizzes, help students learn and teachers monitor student progress [117, 130, 11, 20, 18]. Summative assessments, such as final exams, measure student achievement and provide certification for broader audiences (e.g., parents, school administrators) [130, 117]. Together, effective assessments facilitate learning and teaching [12, 11, 8, 145, 54, 20, 17, 41], and hold all shareholders accountable in the education ecosystem [8, 7, 20, 19, 86, 10, 37].

However, developing assessments is **an extremely time- and knowledge-intensive**

process [94, 140, 78, 124], taking up to half of teachers’ time [126, 125]. Teachers need to create new questions, adapt existing materials, ensure formatting consistency, and align assessments with learning objectives [133, 56, 42]. Yet many teachers receive little training or support for assessment development [124, 125, 131, 56, 86, 42, 95, 41, 131]. Overworked and under-supplied teachers can produce low-quality assessments, creating inaccurate measures of students’ abilities [78, 124, 131, 45]. This exacerbates disparities between high- and low-resource schools, threatening the foundations of educational quality and equity [78, 131, 86].

Prior work has studied how to reduce teachers’ burden of developing assessments by helping them generate individual questions. Automatic question generation (AQG) aims to fully automate question creation with minimal teacher intervention. These methods include template-based generation [144] and more recently LLM-based approaches [39, 90, 89]. While LLM-based approaches have shown promise in generating questions across diverse contexts, they often struggle to produce high-quality questions that can measure higher cognitive abilities [90, 39, 53, 30, 49]. Recent work has studied how to involve teachers in LLM-based approaches to produce high-quality questions, demonstrating the potential of human-LLM collaboration [73, 85]. However, these works mostly focus on generating individual questions and do not provide sufficient insight into teachers’ current assessment authoring workflows. As teachers need to create assessments (rather than individual questions), addressing this gap is essential to build tools that can integrate into their practice.

Our goal is to build a tool that supports assessment authoring through human-LLM collaboration and can seamlessly integrate into teachers’ authoring workflows. To do so, we partnered with 13 teachers from eight U.S. high schools in a three-phase codesign pro-

cess over seven months. In Phase I, we conducted formative interviews to understand teachers' assessment authoring workflows and created a conceptual model of teachers' assessment authoring practices (Section 5.4.1). The model describes assessment authoring as an iterative dual process in which teachers (1) create, reuse, and adapt content at multiple levels (e.g., question level, assessment level) while also (2) refining their requirements about the content of the assessment. We derived from the model a set of design objectives for helping teachers adapt content, review and structure assessments, ensure quality, and integrate requirements.

In Phase II (Section 5.5), guided by the conceptual model and design objectives, we developed **Ripplet**—a web-based tool for authoring assessments—through two rounds of design iteration. Ripplet supports the iterative process of assessment authoring through a **multilevel reusable interaction** paradigm with LLMs. This paradigm supports creating questions from diverse inputs (e.g., a list of topics, PDF of a curriculum), reusing and adapting content at multiple levels (sub-question, question, assessment, cross-assessment), restructuring assessments, and inferring, tracking, and reapplying teachers' edits. Figure 5.1 shows one of the key features of this paradigm—**multilevel reusable edits**—which support iterative authoring: teachers can edit an entire question or just a part of it either manually or by prompting an LLM. The system infers reusable commands from teachers' edits. These commands can be reapplied to multiple questions or specific parts of questions in any assessment, supporting the multilevel nature of assessment authoring.

Over seven months, we collected feedback from codesign partners' independent adoption of Ripplet and observed how their usage and perceptions evolved (Section 5.6). Seven teachers used Ripplet in their classrooms and administered assessments to their students. Some reported that Ripplet helped them create formative assessments they

Multilevel Reuseable Edits

Teachers manually edit a question to align with intended difficulty levels.

Ripplet infers from this manual edit, and creates an edit command

Teachers can **reuse** commands at **multiple levels**: **sub-question** level (e.g., answer), **question** level, or inside **other assessments**.

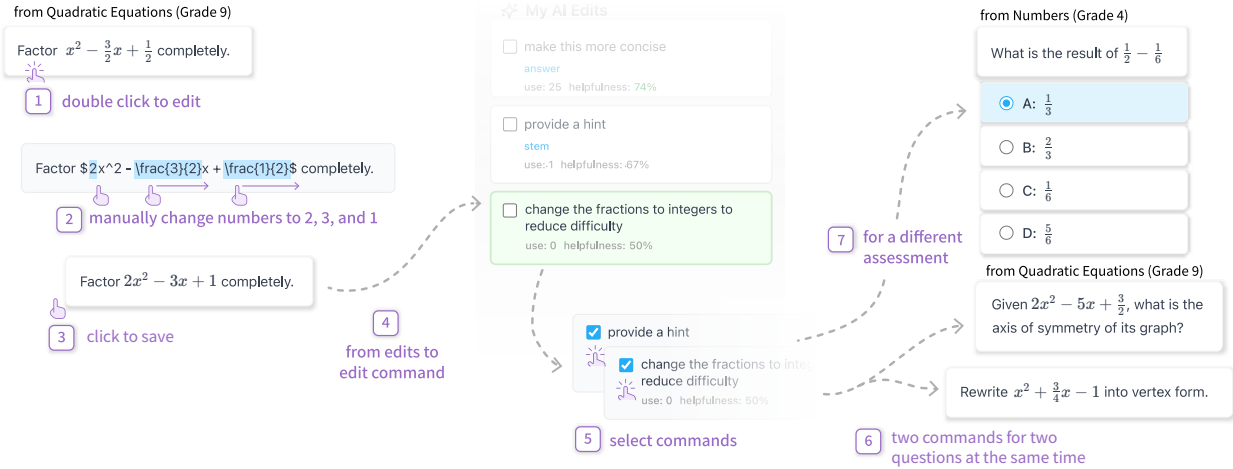


Figure 5.1: An example feature of Ripplet’s multilevel reusable edits with LLMs: A teacher manually edits a question, and then the system infers why this change was made and creates a reusable edit command, which can be reapplied to questions in other assessments.

would not have had time to make, and that they reflected more on assessment design and quality. We also conducted a user study with 15 additional secondary school teachers, comparing their current practices (control) to using Ripplet (Section 5.7). Compared to control, teachers found using ripplet to be a better experience that resulted in higher-quality assessments: enjoyment ($\mu = +2.70$ with 95% CI of $[0.86, 4.54]$), exploration ($\mu = +2.43$ $[0.56, 4.31]$), results worth effort ($\mu = +1.93$ $[0.33, 3.53]$), and assessment quality ($\mu = +1.11$ $[0.10, 2.11]$) all improved on a 10-point scale.

Through this work, we contribute:

- A conceptual model of teachers’ assessment authoring workflows, providing insight into this complex process;
- The design and implementation of Ripplet, a tool for authoring assessments, with

multilevel usable interactions helping teachers to track and apply edits at multiple levels;

- Insight from our codesign sessions, revealing long-term effects of using Ripplet and adoption patterns;
- Findings from a controlled study, showing how Ripplet improves authoring experience and assessment quality.

We also discuss how to build sustainable and reciprocal codesign practices with teachers and how our multilevel reusable interaction paradigm can be applied to support other knowledge-intensive and iterative processes.

5.2 Background

5.2.1 Human–AI Collaboration for Multilevel Iterative Problems

As we discussed in Section 2.3, assessment authoring is inherently **multilevel**: it requires not only generating individual questions but also structuring them into a holistic assessment, tweaking specific parts of a question to fit students’ needs, balancing difficulty and topic coverage, and aligning the collection to curricular goals. As we find in formative interviews (Section 5.4), assessment authoring is also **iterative**: teachers move between creating and adapting questions, searching for relevant materials, reviewing and restructuring the content, and refining their requirements for an assessment.

Researchers in HCI have studied multilevel problems. For example, CoLadder, a code generation system, allows users to construct a tree of prompt blocks, each block representing a different level of abstraction. This multilevel structure allows program-

mers to break down high-level goals into subtasks and procedural details, making the AI-generated code easier to navigate and edit across those levels [150]. In addition, DirectGPT tackles multilevel editing by enabling users to apply changes to a specific part of an object, resulting in less time spent and more preferred outcomes than standard chat interfaces [92]. HCI researchers have also designed techniques to address iterative problems with AI. PromptHive, a prompt authoring interface that supports rapid iteration on prompt variations, helps teachers create hints for educational questions by allowing prompts to be shared and reused [120]. Similarly, ABScribe, an LLM-based system for writing, supports prompt reuse by turning edits into “AI modifiers” that can be reapplied and edited [119]. Recent work by Huang et al. and Zhang et al. demonstrates how LLMs can help solve iterative problems by translating low-level UI interactions into relevant reusable macros and higher-level goals [66, 151]. We draw on these interaction techniques to design a system with multilevel reusable interactions to support assessment authoring.

5.3 Codesign Overview

To design a system that integrates into teachers’ practices, we need to ground our design process in their lived experiences and obtain feedback over an extended period.

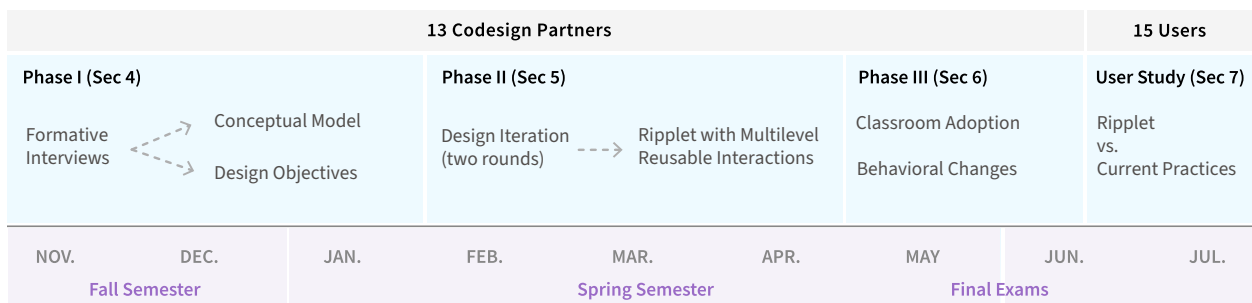


Figure 5.2: Overview of the three-phase co-design process and the controlled user study.

Therefore, we engaged in a **participatory design** process with a diverse group of teachers (Figure 5.2). This process consisted of **three phases** over **seven months**: in Phase I (Section 5.4), we conducted formative interviews to understand teachers’ assessment authoring workflows and challenges. From there, we created a conceptual model of teachers’ authoring practices and defined a set of design objectives for our system. In Phase II (Section 5.5), we conducted two rounds of design iteration where teachers used our prototypes and provided feedback that guided refinements toward the final system. We completed the final version before the next phase. In Phase III (Section 5.6), we invited teachers to independently use Ripplet to create an assessment and administer it to their students at the end of the school year. Afterwards, we conducted follow-up interviews where they reflected on their experience, their students’ reactions, and the overall code-sign process. After codesign, we recruited 15 additional teachers for a controlled user study (Section 5.7).

5.3.1 Participants

We began recruiting by emailing schools that have previous partnerships with our institutions. We then used snowball sampling by asking early recruits to recommend colleagues and friends. We arrived at 13 codesign partners, including 6 female and 7 male teachers from 8 schools (6 public and 2 private) across 3 U.S. states; Table 5.1 describes their characteristics. Everyone was 18 years or older, spoke fluent English, and was actively teaching at the time of the study. The entire codesign was approved by our Institutional Review Board (IRB), and each participant received electronic gift cards as compensation (Phase I: \$40; Phase II: \$60 per session; Phase III: \$90).¹

¹We increased compensation in later rounds to encourage continued participation.

Table 5.1: **Our Codesign Partners' Demographics and Participation.** We have a diverse cohort of partners from a wide range of years of teaching experience, school types, and subjects.

Id	Sex	Years	State	School	Subjects	Formative Inter- view	Design Iter. #1	Design Iter. #2	Indep. Use
P1	F	22	OH	Private	Algebra, Pre-Calculus, Calculus	✓	✓	✓	✓
P2	M	26	OH	Private	History, Geography, Economics	✓	✓	✓	✓
P3	M	28	OH	Private	Biology, Ecology	✓	✓	✓	✓
P4	M	27	IL	Public	Chemistry	✓	✓	✓	
P5	M	23	IL	Public	Physics, Biology	✓		✓	
P6	F	28	IL	Public	Biology	✓	✓	✓	
P7	M	22	IL	Public	Biology	✓		✓	✓
P8	F	9	IL	Public	Biology, Health Careers	✓	✓	✓	✓
P9	M	6	OH	Public	Pre-Calculus, Calculus	✓	✓	✓	✓
P10	F	2	OH	Public	Geometry, Algebra	✓			
P11	F	6	OH	Public	Algebra, Pre-Calculus	✓		✓	
P12	M	11	OH	Public	Algebra, Computer Sci- ence	✓		✓	✓
P13	M	26	NY	Private	History	✓	✓	✓	

5.3.2 General Procedure

All codesign sessions were conducted one-on-one on Zoom. Before the first session, we obtained consent to record both the audio and video and provided them with the option to turn off their video. To facilitate conversations during these sessions, we used seed questions² that probed their practices while leaving space for them to raise questions.

Phase I: Formative Interviews. Each partner participated in an hour-long semi-structured interview to share their assessment authoring practices and challenges. We transcribed all interviews and conducted a thematic analysis [16] on all transcripts. The result is our conceptual model in Section 5.4.1 and design objectives in Section 5.4.2.

Phase II: Design Iteration. After developing the initial prototype of Ripplet, we invited

²The seed questions for each phase are provided in the OSF repository .

our partners to participate in two rounds of design iteration sessions. Each session began with a brief tutorial—covering the entire system in the first round and new functionalities in the second—and was followed by hands-on use of Ripplet while thinking aloud. At the end of each session, teachers participated in a semi-structured interview reflecting on their experiences, likes, confusions, and suggestions for refinement. During these sessions, we took notes on what participants did while using the system, the issues they verbalized while thinking aloud, and their responses during the interview. Over three months, we continually iterated on the design of Ripplet between each codesign session using our observations of teachers' usage and their direct feedback. The result of Phase II is the final version of Ripplet (Section 5.5).

Phase III: Classroom Adoption. Towards the end of the academic year, we invited our partners to use Ripplet independently to create an assessment they would later administer to their students as a practice exam or as the official final exam. After they completed this task, we conducted follow-up interviews in which they reflected on their independent use, their students' feedback, and their overall experiences with Ripplet across the codesign process. A thematic analysis of the interviews gave us insight into how teachers used the system in authentic classroom settings (Section 5.6.1), and also allowed us to capture longitudinal observations on how our codesign partners interacted with the system over an extended period of time (Section 5.6.2).

5.4 Codesign Phase I: Conceptual Model and Design

Objectives

We conducted formative interviews with our codesign partners. From there, we developed a conceptual model of their authoring practices and derived design objectives for

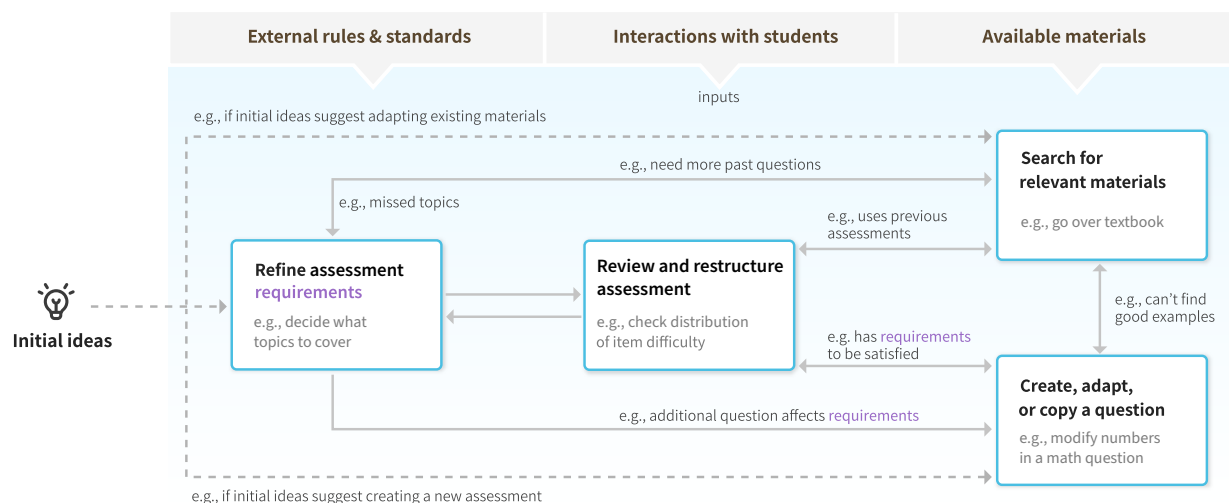


Figure 5.3: **Our conceptual model for teachers' processes of assessment authoring.** In this model, a teacher holds some initial ideas about the assessment requirements. They then start and move between the *stages* to simultaneously develop the assessment and refine assessment requirements, while considering various *inputs*.

our target system.

5.4.1 Conceptual Model for Assessment Authoring

Our conceptual model shows that assessment authoring is an iterative dual process (Figure 5.3) in which teachers (1) **develop the assessment** while they simultaneously (2) refine its **requirements**. In this process, teachers juggle three categories of *inputs* while moving between four major *stages* of action.

5.4.1.1 Inputs

We identify three categories of *inputs* to assessment authoring: (1) external rules and standards, (2) interactions with current students, and (3) available materials.

External Rules & Standards. Teachers do not have complete autonomy in designing assessments; they must adhere to rules and standards from curricula, schools, districts, or

states. In states like Illinois, the Next Generation Science Standards (NGSS) define what students should know and how they demonstrate understanding in K–12 science education [143]. For example, P8 created biology tests emphasizing data analysis to align with NGSS’s focus on cross-cutting skills. Standardized, high-stakes exams also exert a strong influence: Advanced Placement (AP) is a program that offers university-level curricula and exams to high school students in the U.S. and Canada [13]. Teachers often mimic the AP exam formats in their tests <P1, 3, 4, 9>. As P4 noted, *“we do try to mirror the AP exam ... that exam will have some multiple choice and some free response.”* Beyond standardized content, logistical constraints such as limited exam time and quick grading requirements further restrict assessment development: *“I’ve gotta turn grades around within 24 hours, so I can’t do a bunch of open-ended questions ... I have to write a multiple-choice test”* <P5>.

Teacher’s Interactions with Current Students. Teaching is a continuous interaction in which teachers observe students’ progress through classroom activities and formative assessments (e.g., quizzes), then adapt assessments to suit students’ specific strengths, struggles, backgrounds, and expectations. As P11 noted, *“I’m spending a lot of my job watching how my kids do the problem ... when I make the tests, I know I wanna ask the type of questions that they’ve been tripped up on ... I wanna see that they’re getting past that.”*

Available Materials. When creating assessments, teachers typically draw from various resources, including self-curated question banks, colleague-shared materials, official curricula (e.g., released past AP exams), and online sources (e.g., teacher forums). While these materials often prove useful, their quantity, quality, and relevance vary, influencing how much teachers need to adapt or create new content. As P5 noted, *“I’m not sure where they’re sourcing their questions ... their diagrams are really outdated ... I’ll have like an idea of what question I want to ask, and then I’ll go to Google Images ... then I will just re-tailor*

the question to fit that diagram.”

5.4.1.2 Actions

Teachers rarely define all requirements—topics to cover, difficulty levels, question formats, test length, and alignment with *students’ experiences* <P1, 9, 12, 13>—at the outset. Instead, they engage in an **iterative dual** process: they (1) develop the assessment while they simultaneously (2) refine its requirements. Some begin with a vague sense of test topics, refining those topics as they explore *available materials* <P1, 2, 12>. Others draw on semi-developed requirements from past courses and focus on creating or adapting questions, making occasional adjustments to their requirements for current *students* <P8, 9>. We split this dual process into four stages: one for refining requirements, and three for developing the assessment.

Refine Assessment Requirements. Teachers begin the authoring process with a set of initial ideas about their requirements, which evolve as they develop the assessment in the dual process outlined above. Updating the requirements may prompt teachers to *search* from *available resources* or *create* new questions, and to *review and restructure* the assessment for any gaps. For instance, after reviewing past exam performance, P8 “*either change[s] the questions or [gives] students a little bit more support and scaffolding*” to improve a new test. Teachers continue refining requirements throughout the process until the final assessment meets their goals.

Search for Relevant Materials. If teachers’ initial ideas point them to start by adapting *available materials*, they may begin by searching for relevant content, such as course topics, past tests, homework, textbook problem sets, curriculum-prescribed question banks, or online resources: “*I can look through [old textbooks] to quickly glance and see ... that looks*

like it matches what I'm trying to assess from my students" <P1>. From here, teachers may transition to create, adapt, or copy—adapting or copying if they find relevant materials or creating a new question if they do not. In this stage, teachers may also spot gaps that prompt them to review and restructure the assessment or refine requirements: *"I kind of look through [my notes and grade book] and say, am I forgetting anything? Is there anything that we taught that we spent time on that I'd like to test these students on?"* <P12>. Teachers often return to the search stage repeatedly throughout the authoring process to seek relevant materials.

Create, Adapt, or Copy a Question. Teachers may also begin by creating new questions if their initial ideas suggest so. This is common for first-time teachers who may not have sufficient available materials to adapt <P2, 12> or those who prefer novelty and worry about test leakage³ <P4, 13>. For instance, P13 tried to create new questions that differ from students' expectations: *"I try to make ... a question that presents itself a little bit differently than exactly what everybody thinks is coming."* Teachers also adapt or copy past questions. This is often less laborious, especially for experienced teachers. If they find a helpful question by searching from their available materials, they can copy it to the work-in-progress assessment or adapt it to fit the new one; for example, P7 said: *"I've never given the same test ever ... at the very least I take a previous question and change some variables to it so they can do it in a ... different context."* As teachers add questions to the assessment, they regularly review and restructure it to decide what to do next.

Review & Restructure Assessment. If teachers find a past test from searching available materials or have a work-in-progress assessment, they may review and restructure it based on their current requirements, such as coverage of topics, adherence to

³Test leakage is the release or sharing of test materials that could compromise the fairness and integrity of the test; e.g., if past years' students share tests.

external standards, or characteristics of current students: “I tweak [old tests] every year ... I make sure that I cover the standards and the things that I’ve stressed within that semester” <P9>. If teachers find the current assessment does not satisfy their requirements, they may search for materials or create, adapt, or copy questions to address the gaps: “I pulled up the last couple years [of final exams] ... we got further this year ... So the final exam had to account for a test on cell rests. So we also looked at some cell rest questions that we were gonna add to the final to make it a little beefier there” <P7>. Teachers revisit this stage until all requirements are met, which concludes the authoring process.

5.4.2 Design Objectives

The conceptual model illustrates assessment authoring as a complex process. Our interviews revealed challenges in efficiently using these inputs, completing each stage, and moving fluidly between stages. For example, teachers often struggled to efficiently adapt materials across contexts, or to revise assessments without duplicating effort. Building on our conceptual model, we distill a set of design objectives for an assessment authoring system.

DOI: Enable Efficient Reuse and Adaptation of Assessment Content. Teachers rarely create assessments entirely from scratch; instead, they reuse and adapt content from different types of inputs depending on their requirements—for example, using curriculum standards or targeting specific topics. As they move through the process, their requirements evolve, creating the need for efficient ways to create and adapt questions to match updated requirements. Some teachers may need to fill unmet requirements by creating questions on a missing topic, while others want to create similar questions to an existing one to emphasize a concept. Yet this process is often fragmented and inefficient. For example, P12 explained the repetitive burden of this scattered process: “I’ll pull up the

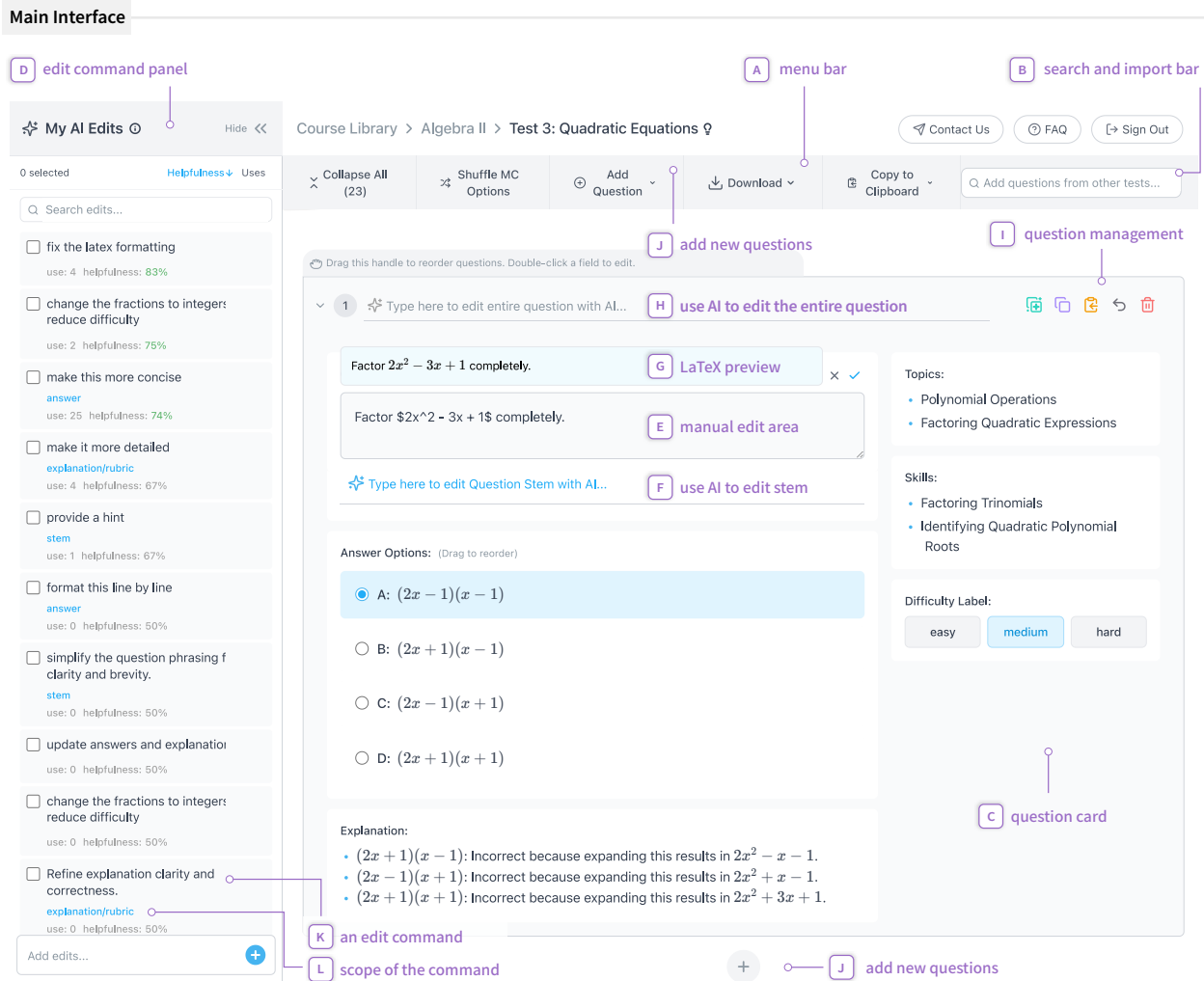


Figure 5.4: The main interface (after double-clicking the question stem) of Ripplet consists of four major components: **A** menu bar, **B** search and import bar, **C** question cards, and **D** edit command panel. The menu bar **A** allows users to add questions and navigate, restructure, and export assessments. The search and import bar **B** allows users to look for questions from other assessments in Ripplet and add to the current assessment. Below **A** and **B** lives the questions in the assessment. Ripplet supports two formats of questions: multiple-choice and free-response. Each question is displayed as a question card **C** which has six sections: question stem, answer options (multiple-choice) / answer (free-response), explanation (multiple-choice) / suggested rubric (free-response), topics, skills, and difficulty label. Users can edit a part of a question manually **E** or with AI **F**. A preview window **G** is shown for content that contains LaTeX . User can also edit the entire question with AI **H**. In addition, on the top right of each question card sit five icons **I** that allow users to manage the question and create additional questions: using AI to generate similar questions, duplicating, copying to clipboard, undoing an edit, and deletion. Users can also add questions **J**. On the left side is the edit command panel **D**. This panel stores the edit commands **K** a user has made to the questions. Each command has a scope tag **L** that specifies the part of the question it will change once applied.

last couple years [of exams] ...and then try to piece them together, but it's a lot of cutting and pasting." Teachers often have to search through multiple documents, reformat content by hand, or recreate questions when small adjustments would have sufficed. A system should not only make it easy to surface content from existing sources but also help teachers adapt that material.

DO2: Facilitate Assessment Evaluation and Restructuring. Teachers emphasized that it is critical to review and restructure an assessment to satisfy their requirements, yet this is cumbersome in existing tools. Teachers working in Google Docs or Word reported spending a large amount of time copying and pasting questions to reorder them. Some also expressed the difficulty of randomizing options for correct answers in multiple-choice questions on a test. A system should help teachers evaluate and restructure assessments fluidly without tedious copy-pasting.

DO3: Ensure Assessment Quality. Teachers emphasized that maintaining the quality of assessments is essential but challenging. Many noted problems with existing materials: for example, P4 described how question banks often contain incorrect and low-quality questions, forcing him to solve problems himself to identify usable questions. Those who experimented with AI tools such as ChatGPT voiced similar concerns, pointing out that hallucinations and unclear phrasing made it necessary to carefully verify correctness <P3–5, 9–12>. A system should help teachers review and validate questions to ensure assessment quality.

DO4: Integrate Requirements into Authoring Workflow. Teachers described requirements as central to assessment authoring, yet in their current practices these requirements are rarely embedded directly into the authoring workflow. Instead, they often exist as separate artifacts—such as checklists, outlines, or mental notes—that must be

manually cross-referenced during assessment authoring. Some requirements are relatively explicit and straightforward to track, such as ensuring balanced coverage of topics across a test. For instance, P5 described printing out his syllabus and marking off topics one by one to verify coverage. Other requirements are more implicit and harder to articulate directly, such as ensuring that questions are written at an appropriate reading level for students <P8>. Meeting these implicit requirements often demands significant additional effort and subjective judgment. This separation makes it easy for requirements to be overlooked and adds extra overhead to an already-demanding process. A system should provide lightweight support to ensure that an assessment satisfies a teacher's requirements.

5.4.3 Challenges and Opportunities for AI Support

While these design objectives are not specific to AI, recent advances in AI—particularly large language models (LLMs)—offer opportunities to help teachers achieve them. For example, prior work has shown that LLMs can generate assessment questions given text or image input, which can help teachers reuse assessment content (**DO1**). Our partners also experimented with tools like ChatGPT to draft or edit questions. However, teachers also encountered clear limitations with existing AI solutions. For instance, many found chat-based interfaces cumbersome for making targeted edits (**DO1**), often scrolling back and forth to locate specific questions or switching between multiple tools (e.g., Google Docs, ChatGPT) to adapt, label, or remove content <P1, 9, 10, 12>. The same limitation also creates a barrier for checking hallucinations and ensuring the quality of assessments (**DO3**). These challenges demonstrate that AI alone is not enough to achieve these design objectives. Instead, it requires carefully designing human–AI interactions so that AI functions are embedded in workflows that reflect real authoring practices and provide support for

teachers to evaluate and refine outputs to ensure assessment quality.

5.5 Codesign Phase II: System Description of Ripplet

Guided by the design objectives in Section 5.4.2, we developed Ripplet, a system that leverages LLMs to support the multi-stage and multi-input process of assessment authoring. We engaged in two rounds of design iteration with our partners and used their feedback to refine prototypes and arrive at the final system. As we refined the design, we converged upon a **multilevel reusable interaction paradigm**, capable of generating questions from diverse inputs, adapting content at multiple levels, restructuring artifacts, and inferring, tracking, and reapplying edits. This paradigm supports question creation (Section 5.5.1), basic question-level (Section 5.5.2) and assessment-level (Section 5.5.3) operations, multilevel reusable edits (Section 5.5.4), and cross-assessment operations (Section 5.5.5).

In this section, we describe the main interface, the operations at different levels, and major refinements (🔄) from design iteration. The main interface consists of four major components (Figure 5.4): **A** Menu Bar, **B** Search and Import Bar, **C** Question Cards, and **D** Edit Command Panel. A demo video of the system is provided in supplemental materials.

5.5.1 Question Creation

To help teachers create questions by reusing content (**DO1**), Ripplet offers five methods for creating and importing questions. Figure 5.5 shows the options from  in the main interface (Figure 5.4): teachers can (1)  write questions by hand, or use LLMs to (2)  generate questions from topics and (3)  curriculum guides, and (4)  import past assessments. Teachers can also (5)  generate variations of an existing question on that question card.  This method was added after the first round of design iteration, when P1, 4, 6, 9 expressed the need to create variations of existing, known-high-quality questions.

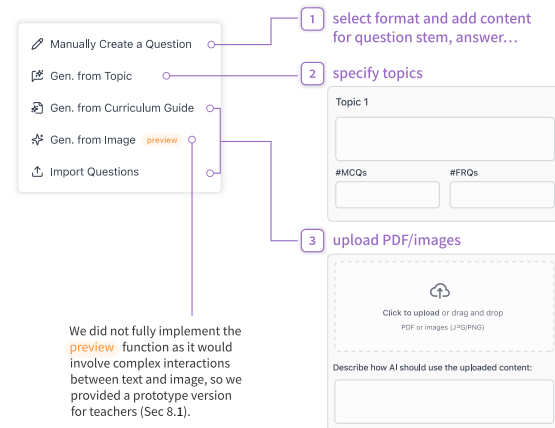


Figure 5.5: The four methods of creating questions with . The preview option provides preliminary support to generate image-based questions, which we discuss in Section 6.1.

5.5.2 Question-Level Operations

Ripplet offers a range of question-level operations that allow teachers to edit and manage individual questions.

Manual Edit. Double-clicking on a part (e.g., question stem) of a question triggers an inline editor for users to make direct, manual changes (Figure 5.4 ). This supports direct edits without requiring navigation away from the main interface, allowing teachers to adapt questions quickly (**DO1**).

Question Management and Version Control. Ripplet provides direct actions on each question card for managing that question (Figure 5.4 ).  Duplication and  undoing changes help teachers efficiently reuse and adapt content across different versions (**D01**), while  deleting allows them to remove weak or irrelevant questions to maintain assessment quality (**D03**).  Copying an individual question to the clipboard allows teachers to export preferred versions and repurpose them in new contexts. These operations enable users to explore question variations while maintaining control over their work, preserving preferred versions, and supporting iterative refinement.

5.5.3 Assessment-Level Operations

Beyond individual questions, Ripplet supports operations that apply to an entire assessment (    ), helping teachers efficiently adapt, organize, and distribute their assessments (Figure 5.4 .

Restructuring. To help restructure assessments (**D02**), Ripplet allows teachers to manually reorder questions via drag and drop.  Shuffling MC options randomizes the order of options within each multiple-choice question, helping reduce answer pattern bias and quickly create alternative versions of an assessment.

Overview and Navigation. Ripplet provides features to help teachers gain an overview of assessments and navigate them efficiently. Hovering over  next to the assessment name displays the composition of questions by format and difficulty in that assessment, giving an at-a-glance summary to help teachers understand what requirements still need to be satisfied (**D04**). Teachers can  collapse or expand all question cards at once, making it easier to move through large assessments during evaluation and restructuring (**D02**).

Exporting. Ripplet provides two export options:  download and  copy to clipboard.  Downloading generates a formatted PDF of an assessment with or without answer keys and explanations/suggested rubrics. Teachers can also  copy the assessment to the clipboard and then paste it into other software (e.g., Word or Google Docs) to add finishing touches (e.g., adjusting fonts) before distribution. Expressions, equations, and special symbols are preserved as inline images on the clipboard to ensure that they display as expected when pasted in other software.

5.5.4 Multilevel Reusable Edits with LLMs

In addition to the basic operations at the question and assessment levels, Ripplet provides multilevel reusable edits which allow teachers to edit and reuse edits at the sub-question, question, and assessment level.

Multilevel Edit. At the sub-question level, double-clicking on a specific part of a question allows teachers to prompt an LLM to make an edit to that part of the question (Figure 5.6A; **DO1**). To show the LLM-edited version, Ripplet uses an inline difference view that highlights modifications, enabling a focused review of change and quality (**DO3**). Teachers can then accept or reject the change. This allows teachers to adapt questions without manually editing content.  This sub-question level editing feature was implemented after the first round of design iteration, where we noticed that many partners wanted to edit only specific parts of the question to prevent unwanted changes in other parts.

At the question level, teachers can also edit the entire question by entering instructions in the prompt bar at the top of the question (Figure 5.6B). Once the LLM completes the request, Ripplet displays a side-by-side comparison of the original and LLM-edited versions so that teachers can easily review the change and question quality (**DO3**) and de-

cide whether to accept or reject the change. All LLM edit requests are added to the edit command panel for later reuse.

Inferred Reusable Edit. Whenever a user makes a manual edit, the system sends the previous and updated versions of the question to an LLM and requests it to infer why this change was made and generate a generalized edit command on the command panel that teachers can later apply to other questions. In Figure 5.1, for example, a math teacher edits a question stem to change fractions to integers. The system infers the underlying requirement (*change fractions to integers to reduce difficulty*) and adds it to the command panel (**DO4**).

Multilevel Reusable Edits. The edit command panel (Figure 5.4 ) tracks the edits a teacher has made and allows them to apply edits to multiple questions simultaneously, supporting efficient question adaptation (**DO1**). This panel automatically stores the multilevel edits a teacher has made and the inferred edit commands from manual edits. Teachers can also create new edit commands from scratch by typing into the text box at the bottom of the panel. Each command has two metrics: *uses* (the number of times the edit has been applied) and *helpfulness* (calculated by updating a beta-distributed prior on helpfulness with the counts of accepted and rejected edits). The panel supports sorting by either metric or keyword search to quickly locate commands.

To apply a command, teachers select it from the panel and choose one or more target questions (Figure 5.7). A comparison modal (right side of Figure 5.7) allows the teacher to toggle between questions and accept or reject the LLM-generated version individually to ensure question quality (**DO3**).

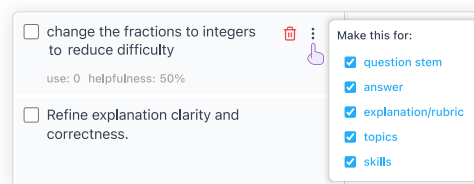


Figure 5.8: Each command can be tagged with specific parts of a question to ensure that only those parts are modified when the command is applied.

Initially, we only stored the text of each edit command without storing which part of the question a teacher was editing when that command was created. This meant that the entire question could be changed if a command was reused on another question. For example, if a teacher edited a question stem by asking to “make this shorter” then reused the resulting command from the command panel on another question, it would attempt to make every part of that question shorter. During iteration, we observed that teachers wanted to control which parts of a question to edit when applying a command from the panel.  We added adjustable tags to commands that specify which parts of a question (e.g., question stem, answer, topics) it should apply to (Figure 5.8).

Reusable Commands As a Way to Enact Requirements. The reusable commands not only help teachers adapt assessments, but also better integrate teachers’ implicit and explicit requirements into assessment authoring by tracking and surfacing their edits (D04). Initially, we experimented with designs where requirements are reified in the main interface, e.g. as a nested checklist. We imagined the system could help teachers write down and mark off items on that list by identifying which question(s) contribute to each requirement. We quickly realized a reified requirement checklist does not fit the implicit and entangled nature of assessment requirements. Some requirements are difficult to articulate up front, such as using only positive numbers in a question about factoring so that students can focus specifically on factoring. Other requirements are entangled: for example, when teachers adjust question difficulty, they might simultaneously change the coverage and progression of topics in the assessment. It would be difficult to isolate and articulate all such requirements explicitly. However, we noticed that teachers often discover requirements as they are working on specific questions. We designed multi-

level reusable edits to capitalize on this behavior by eliciting, tracking, and surfacing a teacher’s requirements implicitly from the edits they make, without imposing the burden of writing down explicit requirements.

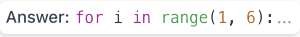
Managing the Command Panel with Similarity Checker. Teachers often make many edits to an assessment, either manually or with the help of LLMs. During design iteration, this could generate a large number of edit commands and quickly overwhelm the command panel, reducing its usefulness. While teachers could use the search and delete functions to manage the panel, this requires additional effort. However, many edits are similar. For example, P3 repeatedly removed extra words from question stems to keep them concise, and P12 repeatedly used AI edits like “add a hint” or “make the question easier by giving additional information”.  To reduce the list of edit commands on the command panel, we implemented an LLM-based similarity checker. When a new command is generated, either directly from an AI edit request or inferred from a manual edit, it is passed to an LLM along with the current list of commands on the panel to judge if the new command is sufficiently different from all existing commands. We collected common examples of similar commands and used few-shot prompting for the checker’s prompt template to help define command similarity. The similarity checker significantly reduced the number of repeated commands on the panel and made the panel more usable for teachers.

5.5.5 Cross-Assessment Operations and Supports

Ripplet also provides tools that operate across assessments to help teachers reuse and adapt materials (**DO1**). These capabilities include making edit commands available across assessments, importing questions from other assessments, and support for special expressions and equations and syntax highlighting for code.

Sharing Edit Commands across Assessments. Initially, the edit commands were only preserved within an assessment. Multiple partners, including P1 and P12 who used Ripplet extensively, indicated that they needed to apply similar edit commands across assessments and even across courses. Therefore,  we made each teacher's edit commands available across their assessments to help them adapt other assessments without re-specifying commands (DO1).

Importing Questions from Other Assessments. To support reusing high-quality content (DO1), Ripplet allows teachers to add questions from previous assessments via the search and import bar (Figure 5.4 ). Teachers can search and filter past assessment questions by a variety of criteria, such as class, topic, format, and difficulty (Figure 5.9), then view full question details and directly insert questions into the current assessment.

Equation Authoring and Syntax Highlighting. In our formative interviews, mathematics and science teachers described significant challenges in creating, editing, and formatting expressions and equations using existing tools like Microsoft Word. To display these expressions properly, Ripplet enables users to render and author expressions and equations using $\text{\LaTeX}$ notation. Recognizing that teachers typically lack familiarity with $\text{\LaTeX}$, we automatically display a rendered preview tab whenever mathematical expressions or markdown blocks are detected (e.g., $\$ \dots \$$ or $\backslash\text{begin}\{\text{equation}\}$) (Figure 5.4 ). To support computer science teachers, we implement syntax highlighting for all major programming languages (e.g., Python, R, Rust) using the same tab interface for displaying rendered results. () All rendered content—both mathematical and scientific equations and highlighted code—can be exported to PDF or copied to the clipboard for integration into other software.

5.5.6 Ripplet Walk-through

To demonstrate the Ripplet workflow and how its multilevel reusable interactions support assessment authoring, we walk through a fictional scenario where a teacher, Ms. Ripley, creates an assessment (Figure 5.10).

Ms. Ripley is teaching an Algebra II course that has a test coming up on linear equations. She logs in to Ripplet, enters her course library and clicks on her Algebra II course. She enters the assessment library and begins creating her test. Ms. Ripley sees several ways to do this and chooses “Generate from Topic” because she has specific topics in mind: *solving equations in slope-intercept form* and *converting them to standard form*. She also has a rough idea of how many multiple-choice and free-response questions a balanced test should have to take up the full class period.

After she clicks “Generate”, Ms. Ripley enters the main interface with the assessment. After quickly *reviewing* the questions, she decides to *add additional questions to cover a missing topic*. This opens a modal with the same question creation methods she saw before. This time she *writes a new question* on linear equations from scratch. Now that her test satisfies her length and topic coverage requirements, Ms. Ripley carefully *reviews* and *revises each question*. She notices that the suggested rubric for a free-response question is not detailed enough, so she double clicks on it and asks AI to provide more detail. She *checks the AI’s output* and notices that the new rubric is too detailed, so she rejects the change and edits it manually. She then notices that some questions may be *too hard for her students* because the solutions involve fractions, which her students have been struggling with. For this test, she only wants to assess students’ ability to work with equations of a line, not their understanding of fractions, so she reuses the command “change the fractions to integers to reduce difficulty” in the command panel and applies it

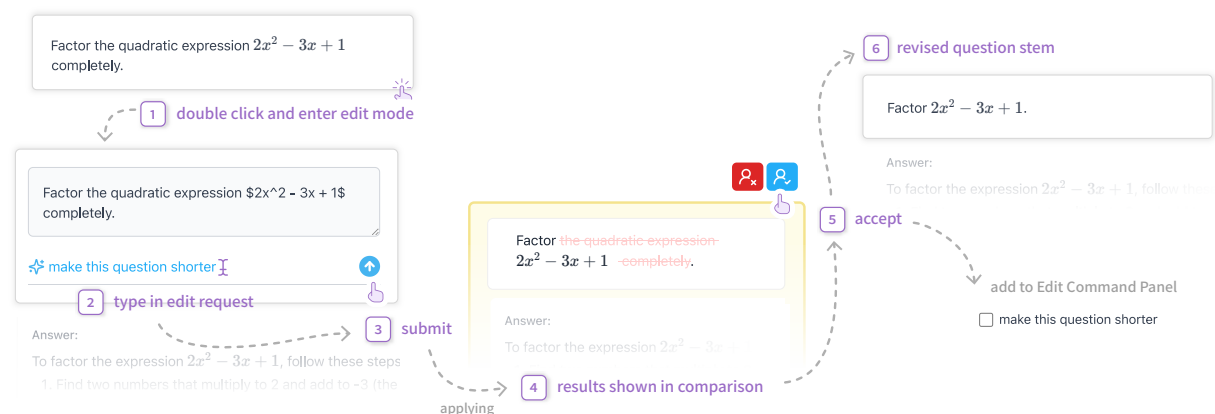
to all corresponding questions at once. She reviews the changes and accepts the suitable ones. Once satisfied, she downloads it as a PDF to hand out in class.

5.5.7 System Implementation

We built Ripplet using TypeScript with React for the frontend, PostgreSQL for the database, and a Python backend with Flask. We use OpenAI’s GPT-4o API for LLM-related features because it supports structured output, which allows the system to receive well-formed JSON responses that can be directly parsed and integrated into the application logic with minimal post-processing. We used GPT-4o with vision for generating questions from curriculum guides and importing questions, as it can extract special symbols and expressions from images of documents. The prompt templates of some LLM-based features benefited from collecting examples from codesign partners. For example, we used common examples of similar commands to revise the prompt template of the similarity checker.

In the database, we store the full edit history of questions for version control. We also store the full history of edit command usage to calculate the helpfulness and use metrics. In the interface, we render mathematical equations for in-browser display and copy-to-clipboard using `remark-math` [116] and `rehype-katex` [115], while for PDF rendering we use `KaTeX` [74]. Similarly, for syntax highlighting, we use `react-syntax-highlighter` [114] for in-browser display and copy, while `prismjs` [112] is used for rendering PDFs.

A. Multilevel Edit (sub-question level)



B. Multilevel Edit (question level)

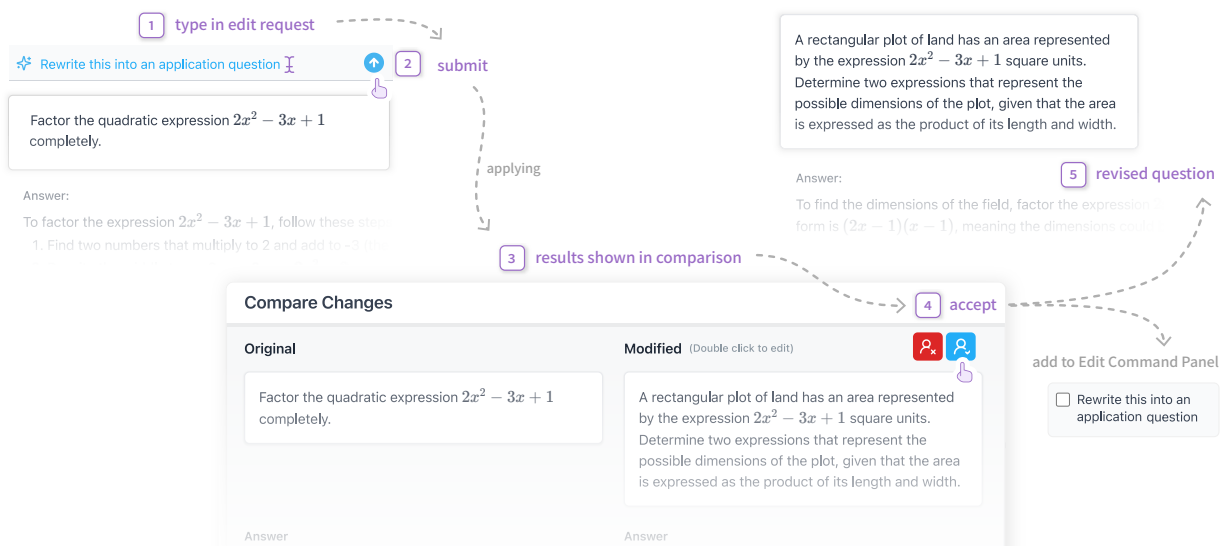


Figure 5.6: Multilevel edit in Ripplet. Teachers can revise either a specific part (top) or the entire question (bottom) with AI. Differences are shown inline or side-by-side for quick review.

Multilevel Reuseable Edit (command panel)

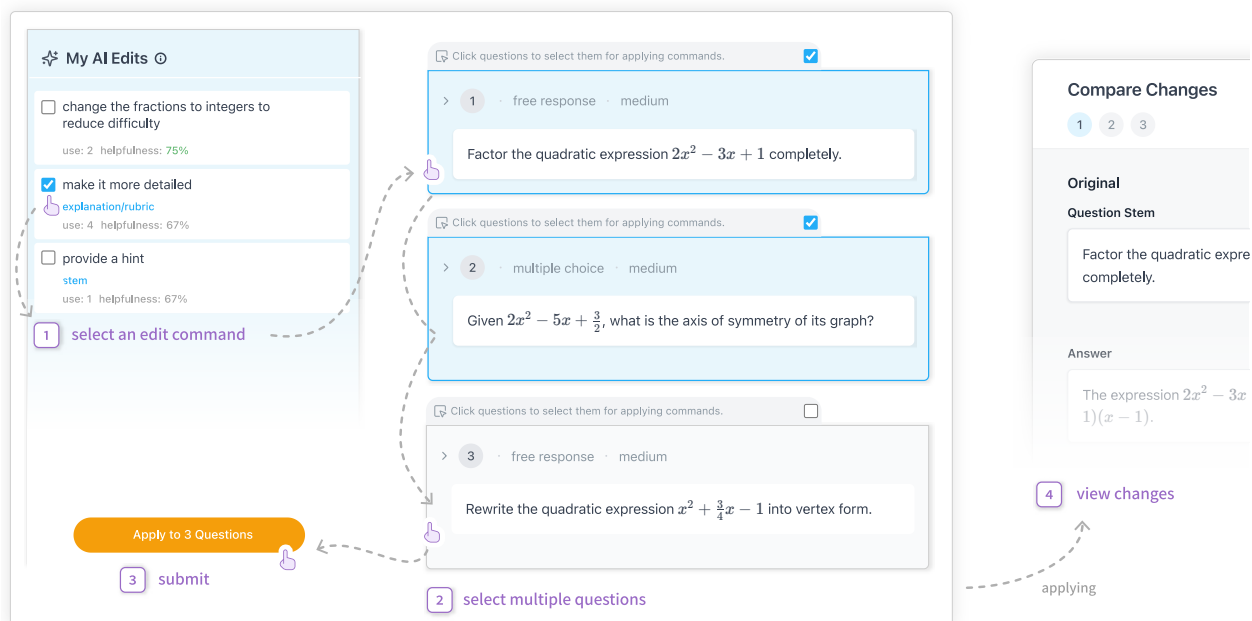


Figure 5.7: Multilevel reusable edits from the command panel. Teachers can select a command and apply it to specific questions. They can review the AI-generated changes with the original question side by side. Tabs allow switching between questions, and each change can be accepted or rejected individually.

Search and Import Bar

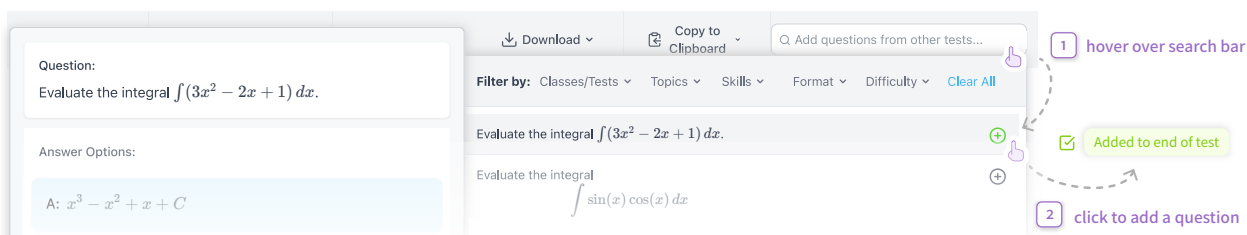


Figure 5.9: Search and filter interface for reusing questions from other assessments. Teachers can search by class, topic, format, and difficulty, and preview questions before adding them.

System Walkthrough

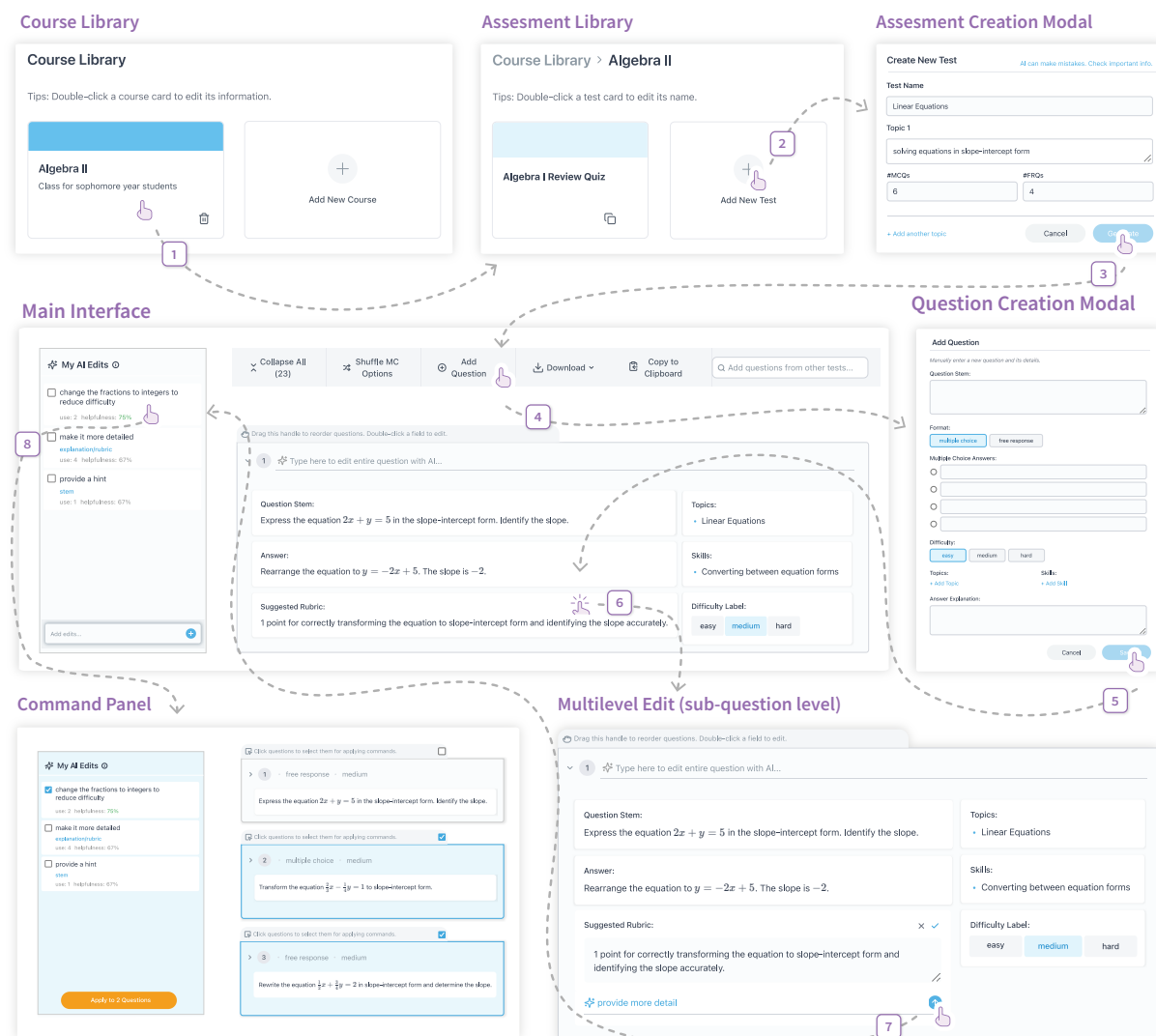


Figure 5.10: **System walkthrough of Ripplet.** Ms. Ripley begins in the course library, navigates into the assessment library, and creates a new assessment through the test creation modal by entering topics and specific number of questions in each question format. She then reviews the assessment in the main interface and writes a question manually. She then uses AI to edit the suggested rubric of a question so that it has more detail. Finally, she notices that several questions may be too hard for her students, so she reuses the command “change the fractions to integers to reduce difficulty” in the command panel and applies it to all corresponding questions at once. She reviews the revisions and accepts the suitable changes. Once satisfied with the test, Ms. Ripley exports it as a PDF.

5.6 Codesign Phase III: Independent Use and Longitudinal Observations

Our codesign process provided a unique opportunity to track Ripplet’s use across two complementary dimensions: independent adoption and longitudinal engagement. Seven teachers used Ripplet independently to create and administer assessments in their classrooms, giving us insight into how the system functioned in organic settings. In addition, because many of the same teachers participated in multiple rounds of codesign sessions and continued corresponding between sessions, we were able to observe how their practices and preferences evolved over several months.

5.6.1 Independent Classroom Adoption

Following the second round of design iteration, seven teachers used Ripplet to create assessments and administered them to their students. They created a variety of assessments—from official final exams for state-required classes to practice tests for dual-credit community college courses.

Ripplet Integrated into Teachers’ Existing Workflows. Teachers’ independent use revealed that Ripplet fits naturally into a variety of existing workflows by supporting different inputs and stages. For example, P9 imported high-quality AP questions and used the “Generate Similar” and AI editing features to create a review packet, while P8 uploaded curriculum standards to generate a mock exam for her students in the health careers course and used a mix of manual editing and the command panel to reduce the reading level of some questions. P1 started with topic-based generation to target specific areas for her students; some questions inspired and reminded her to cover a different topic

and blend multiple topics together into one question, so she created new ones both manually and with AI. Finally, she reordered the questions so that they progress in difficulty. By supporting diverse question generation, adapting questions at multiple levels, and restructuring assessments, Ripplet allowed teachers to build on their materials and navigate the iterative authoring process without abandoning their established ways of working.

Time Savings Enabled Teachers to Create Materials They Could Not Before. Teachers expressed that Ripplet enabled them to create assessments they previously lacked the time or resources to produce. P8, for instance, generated a 28-question practice final exam in just 30 minutes—a task that would normally have taken her over five hours. She emphasized that the efficiency made the creation of more formative assessments possible: *“I just wouldn’t have made it otherwise.”* Similarly, P9 explained that without Ripplet, she would have had to purchase a commercial review packet, but instead she was able to build one herself quickly and give it to her students. Teachers noted that these new assessments had clear downstream benefits for students: P8’s class treated the mock exam as seriously as their actual final and learned a lot by working through the mock exam, and P9’s students reported that the review packet was helpful in preparing for the final exam. These accounts underscore Ripplet’s educational impact of not only saving teachers’ time but also expanding the opportunities they could offer their students.

Multilevel Organization Eased Assessment Refinement. Teachers also praised how Ripplet’s structured interface supported the stages of assessment authoring, particularly restructuring assessments and revising questions. The system’s card-based question display and menu bar helped them reorder questions and shuffle multiple-choice options—tasks that were cumbersome in tools like Google Docs, which required manually cutting and pasting text. P12 described the organizational structure as *“the huge benefit with Rip-*

plet ... each question is its own element, and it's easy to make localized changes." Being able to edit specific parts of questions with AI further supported this process by allowing teachers to refine multiple-choice distractors or rephrase questions stems without disturbing the other parts of questions <P3>. The multilevel organization gives teachers more control over how their assessments were assembled, supporting evaluation and restructuring while reducing the tedium of manual formatting.

5.6.2 Behavioral Changes in Assessment Authoring

Evolving Ownership: from Generation to Curation. As teachers engaged with Ripplet over time, their sense of ownership and role in assessment design shifted. Early in the codesign process, our partners spent more time generating questions with LLM-based methods. While they emphasized the need to reuse and adapt prior materials, they did not utilize many features designed for that purpose (e.g., importing assessments, command panel). As a result, many perceived AI as the “real” author and themselves as proofreaders, often questioning whether they could truly claim ownership of the assessments. With repeated use, however, their behavior and perception evolved. Teachers routinely imported questions from their own past tests, modified them for new contexts, and wove them into new assessments. P12 highlighted the efficiency of being able to “*add questions from other tests and filter them.*” Similarly, the command panel for reusing edit commands—initially underexplored—grew into a valued tool. In the first round, some overlooked the command panel entirely; by the second, they praised it as a timesaver that improved productivity <P4, 8>. Teachers also grew comfortable deleting low-quality questions, rejecting weak AI suggestions, and refining assessments manually. Teachers increasingly recognized themselves as the primary creators. P9 explained that at first she felt “*the AI did most of the work,*” but later came to see that she was “*the one deciding what to keep,*

what to change, and what made sense for my students.” In addition to teachers’ growing familiarity with AI and their development of more effective prompting strategies, several features of the system likely contributed to this transition, such as the side-by-side comparison modal and inline difference view of AI editing, which allows teachers to control the quality (D03) of assessments by reviewing, accepting, or rejecting AI edits easily.

Becoming More Reflective by Writing and Seeing Commands. Teachers also reported that Ripplet made them think more deliberately about assessment design and question quality. P1 explained that using AI to adapt questions forces her to think more critically about what it is that makes a question difficult. Likewise, P3 reflected that his usage of Ripplet encouraged him to slow down and become “*more deliberate in what I do, instead of just going through the motions.*” Similarly, P12 expressed that seeing the commands showing up on the command panel helped him understand his own requirements and reflect on what needed to be done to design a high-quality assessment (D04). By inferring and surfacing teachers’ requirements as reusable edit commands, the system helped teachers approach assessment authoring in a more systematic and reflective way.

Extending Ripplet’s Applications in Teaching. Extended engagement also reshaped how teachers envisioned their professional practice. P8 described wanting to use Ripplet on demand during office hours to generate practice questions, while others imagined using it for multilingual or remedial contexts (Ripplet supports the display of text in multiple languages). Teachers also highlighted time saved outside of assessment authoring: P3 and others noted that Ripplet’s answer keys and explanations allowed them to hand these directly to students, reducing the need for lengthy office hours. P1 explained that students can self-diagnose first instead of her hosting a multi-hour session to explain the answers. These reflections show that teachers came to see Ripplet not only as an assess-

ment authoring tool but also as part of a broader ecosystem of teaching, learning, and collaboration.

5.7 Controlled User Study

While our codesign process provided rich insights into how teachers engaged with Ripplet over time, it involved codesign partners who were deeply familiar with the system. To evaluate Ripplet on a sample of teachers less familiar with the system or invested in its design, we conducted a controlled, within-subjects study with 15 other teachers. Each teacher created assessments in two conditions: once with their current practice (control) and once with Ripplet. This design allows us to make a direct comparison between Ripplet and their current practices, providing evidence about whether Ripplet’s benefits generalize beyond our codesign partners.

5.7.1 Participants

We asked our codesign partners to reach out to their teacher networks through mailing lists and Facebook groups. Table 5.2 shows the 15 teachers (11F, 4M) we recruited from 13 schools in 4 states, using the same criteria as the codesign recruitment (see Section 5.3.1). The group included 13 high school and 2 middle school teachers across different subject areas: mathematics and computer science (6), natural sciences (4), and social studies (5). Their teaching experience ranged from 1 to 31 years. All participants completed a pre-survey that included questions about their experience and comfort with using AI tools. Some had never used AI in their work and self-identified as having “no ability” in using AI <U14>, while others identified as “advanced” users and had used tools such as ChatGPT to create and adapt assessments frequently <U2>. The pre-survey also asks each teacher to

Table 5.2: Information about teachers in controlled user study.

Id	Sex	Years	State	School	Subjects
U1	F	19	OH	Private	Geometry, Pre-Algebra
U2	M	8	OH	Public	Chemistry
U3	F	18	VA	Private	Economics, Human Geography, Business
U4	F	30	OH	Public	Biology, Anatomy
U5	F	1	IL	Public	Biology, Physiology
U6	F	8	TX	Public	Biology
U7	M	31	OH	Public	Middle School History
U8	M	13	TX	Public	Economics, Financial Literacy
U9	F	27	OH	Private	History, Psychology
U10	F	7	OH	Public	Pre-Algebra, Algebra I, Geometry
U11	F	5	OH	Public	Algebra I
U12	F	6	OH	Public	Algebra II, Statistics
U13	F	20	OH	Private	Middle School Math
U14	F	29	OH	Public	Financial Literacy
U15	M	15	IL	Public	Computer Science

list the courses they will teach next semester and two chapters or units in each course that are similar in length and difficulty. This study was IRB-approved, and each participant received a total compensation of \$195.

5.7.2 Procedure

The study involved two Zoom meetings, each followed by an asynchronous task that teachers conducted and recorded themselves. To counterbalance potential order effects, we randomly assigned participants to either a control-first or Ripplet-first group, then randomly picked a course they listed in the pre-survey and randomly assigned two units from that course to the two conditions.

In the control-first group, participants began with a 15-minute Zoom session where we introduced the study and outlined the first asynchronous task. In this task, we asked teachers to create an assessment for a unit of a course that they would teach the next

semester, using their current method of authoring assessments. While working on the task, they recorded their screens and narrated their process by thinking aloud. Afterward, they completed a survey (Section 5.7.3) reflecting on their experience. The second Zoom meeting (60 minutes) began with a 10-minute tutorial video introducing the Ripplet tool followed by 40 minutes of hands-on exploration. During this time, participants experimented with Ripplet and received support from the research team as needed. In the last 10 minutes, we introduced the second asynchronous task: creating an assessment using Ripplet for the unit assigned to the Ripplet condition. As with the first task, the participants recorded their screens, narrated their process, and completed a post-task survey. In the Ripplet-first group, participants followed the same set of activities and tasks, but in reverse order.

Table 5.3: Survey items grouped by five areas. Participants were asked to rate each item on a scale from 1 to 10.

Area	Survey Item: (highly disagree [1] - highly agree [10])
Enjoyment	I would be happy to use this way of creating assessments on a regular basis. I enjoyed using this way of creating assessments.
Exploration	It was easy for me to explore many different ideas, options, designs, or outcomes, using this way of creating assessments. This way of creating assessments was helpful in allowing me to track different ideas, outcomes, or possibilities.
Results Worth Effort	I was satisfied with what I got out of this way of creating assessments. What I was able to produce was worth the effort I had to exert to produce it.
Perceived Control	I felt I had a say in the assessment creation process. I was able to influence the assessment creation process.
Assessment Quality	The assessment I created can measure my students' skills and knowledge or help them improve. The assessment questions are worded clearly. The assessment is adequately difficult for my students.

5.7.3 Survey Design

We asked participants to evaluate Ripplet and their current practices in five areas through an 11-item survey (see Table 5.3). As assessment creation is a creative process, we first adapted the Creativity Support Index (CSI) [32] by selecting the areas applicable to evaluating our system: “Enjoyment”, “Exploration”, and “Results Worth Effort”. Because we need to administer the survey in both the control and Ripplet conditions, we changed phrases of the CSI statements that read “the/this system or tool” to “this way of creating assessments”. To understand users’ sense of agency and control in a system involving LLMs, we measured perceived control with two questions from Lee et al. [80] and adapted them by changing “content editing” to “assessment creation” to suit our setting. Finally, to understand the educational impact of Ripplet, we added three questions regarding the quality of assessments. We adapted two questions from Cui et al. [39] for measuring the relevance and clarity of assessments, and we added another question on the difficulty of the assessments. We also invited five codesign partners for feedback and finalized the survey after they confirmed that the questions capture the most important aspects of assessment quality.

5.7.4 Quantitative Comparison

To compare participants’ authoring experiences and the quality of assessments from using Ripplet vs. their current practices (control), we compared their ratings in each of the five areas on the survey.

Analysis. First, based on the guidance for analyzing Likert items, we averaged ratings for items within each area for each participant and condition [24, 100]. For each area, we then compared Ripplet vs. control using paired-samples *t*-tests on the within-participant

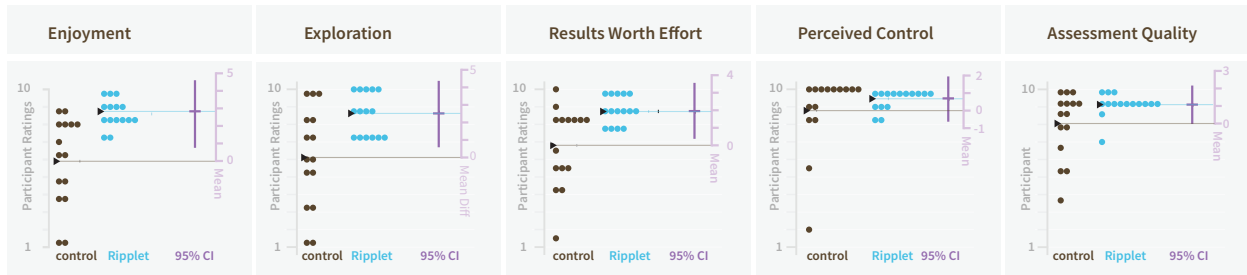


Figure 5.11: Distribution of teachers’ ratings for each of the five areas in both conditions, with mean differences (Ripplet – control) and their 95% CIs from the paired t -tests.

mean differences (Ripplet – control). To account for the five parallel tests, we controlled the false discovery rate using the Benjamini–Hochberg (BH) procedure ($\alpha = 0.05$), and we report the BH-adjusted p value, mean difference, and its 95% CI for each of the five areas. We chose the paired t -test because it is high-power and sufficiently robust to modest assumption violations [100, 127, 147].⁴ This analysis plan was pre-registered on OSF.⁵

Results. Figure 5.11 shows the distributions of participants’ ratings for the five areas in both conditions, as well as the mean and the confidence interval of the difference in ratings (Ripplet – control). **Ripplet shows significant improvement over control in four areas:** enjoyment ($\mu = +2.70$ with 95% CI of [0.86, 4.54], $p = 0.003$), exploration ($\mu = +2.43$ [0.56, 4.31], $p = 0.012$), results worth effort ($\mu = +1.93$ [0.33, 3.53], $p = 0.012$), and assessment quality ($\mu = +1.11$ [0.10, 2.11], $p = 0.032$). Ratings for perceived control are similarly high across conditions, possibly because teachers in both settings could manually create and edit questions.

⁴ t -tests can be performed on low-number-of-item Likert scales because parametric statistics are robust with respect to Likert being ordinal [100].

⁵OSF link: https://osf.io/j8zey/?view_only=63a88f03b3834e8b8aa4b2ed38e4707c

5.7.5 Qualitative Findings

In addition to rating both conditions, participants answered two optional open-ended questions: (1) which features or functionalities they found most helpful, and (2) the difficulties they encountered. Their responses corroborate our observations from our code-sign process (Section 5.6). Participants highlighted that multilevel reusable edits were efficient for revising assessments <U3, 5, 8, 10, 12>. They also valued functions such as generating similar questions, which allowed them to reuse high-quality materials <U2, 13>, and shuffle MC options <U9>, which allowed them to restructure assessments. U8 noted that Ripplet would enable him to create more formative assessments for the upcoming school year—assessments he otherwise would not have had time to prepare.

At the same time, several math teachers mentioned their unfamiliarity with $\text{\LaTeX}$ and often had to ask AI to correct related issues <U11-13>. U13 expressed confusion that similar prompts sometimes produced different results, suggesting that understanding the uncertain nature of LLMs and adjusting expectations could take time. Teachers also expressed a desire for support in generating questions with visual representations <U5, 8>, suggesting an important area for future work on assessment authoring systems (Section 6.1).

5.8 Discussion

5.8.1 Effective Codesign Requires Aligning with Teachers' Schedule and Supporting their Growth

Engaging teachers in a seven-month codesign process across two semesters proved invaluable for developing Ripplet, and it also surfaced lessons about both the opportunities and challenges of sustained collaboration. Many teachers began with limited prior exposure to AI tools. Through the codesign process, they developed a better understanding of the probabilistic nature of these systems and learned strategies to frame prompts and manage expectations. In this way, codesign was not just about providing feedback on Ripplet, but also an opportunity for teachers themselves to build new knowledge and skills around working with AI.

At the same time, sustaining participation was demanding. Longitudinal codesign required repeated follow-ups and building trust and relationships. Teachers' workloads left little time to devote to research activities, and one partner remarked that she wished she had *“more time to play around with the tool outside [of the scheduled Zoom sessions].”* In addition, we timed Phase II to span a semester and Phase III to be when teachers are writing final exams so that teachers could meaningfully test Ripplet during periods when they were actively teaching and developing assessments. It is important to not only create intentional opportunities for open-ended exploration outside structured sessions, but also carefully align system design and evaluation periods with the rhythms of teachers' academic calendars.

We also observed differences in how teachers articulated design feedback. One

partner with prior training in design thinking provided specific, actionable suggestions, while another partner defaulted to broader descriptors such as “user friendly” or “hard to use.” Although all input was valuable, this variation suggests the opportunity for lightweight training to help codesign partners express more concrete, design-oriented feedback without adding extra burden. Building reciprocal, sustainable, and effective codesign practices requires structuring collaborations that honor teachers’ time and rhythms, while also creating space for their growth as designers and learners.

5.8.2 Beyond Assessment Authoring: Applications of Multilevel Reusable Interactions

Ripplet’s multilevel reusable interaction paradigm—generating from diverse inputs, adapting at multiple levels, restructuring artifacts, and inferring, tracking, and reapplying edits—can help scaffold complex assessment authoring workflows. In education, this interaction paradigm has potential applications beyond assessment authoring. For instance, grading and writing feedback often involves recurring edits that could be surfaced and reapplied through similar edit command panels. By capturing these refinements explicitly, systems can make teachers’ tacit strategies visible, reusable, and transferable across tasks.

More broadly, multilevel reusable interactions provide a design direction for domains where human creativity and judgment must be balanced with LLMs’ generative capacity. In product or graphic design, for example, users frequently revise drafts by applying recurring transformations. A command-tracking interface could infer reusable operations from users’ edits and provide metrics for users to understand how helpful they might be. By using LLMs as collaborators that learn from and support human refine-

ments, this paradigm can extend to a wide range of professional and creative practices. As we demonstrated in the codesign evaluation and controlled user study of Ripplelet, this paradigm has the potential to help people become more reflective and deliberate in their practices and more efficiently create high-quality, curated artifacts.

Chapter 6

Discussion

6.1 Iterative Authoring of Questions with Visual Representations Remains Difficult

Visual representations—such as free-body diagrams in physics <P5> or choropleth maps in geography <P2>—are commonly used across disciplines to convey complex scenarios to assess students’ domain knowledge or data analysis skills. Creating them often requires specialized software, but teachers may lack the requisite technical expertise or the budget and time to purchase and learn how to use these tools. Ripplet and VILA provide preliminary support for generating questions in one shot. However, in practice, one-off generation for questions with visual representations is insufficient. Teachers often need to edit the visual representation and the text iteratively to ensure coherence and quality. Recent research has shown that LLM-based solutions, such as VILA, can generate questions with visual representations. Adapting these solutions for classroom practice will require multi-modal interfaces that allow teachers to edit visuals and text together, with systems that track dependencies and suggest aligned revisions. Because questions with visual representations play an important role in assessments and are hard to create,

improving support for this process could have a wide impact.

6.2 Lowering Technical Barriers Is a Prerequisite for Adoption and Impact

While efficiency and quality are central goals of technology for assessment development, these qualities only matter if systems are actually used in practice. Across the three projects, a clear lesson emerges: lowering the technical barriers to access, setup, and ongoing use is critical for making assessment technologies genuinely adoptable.

Even the more accurate or adaptive tools cannot create impact if they remain inaccessible to their intended users. For example, the adaptive visualization literacy assessments (A-VLAT and A-CALVI) demonstrated strong improvement in efficiency, substantially shortening test length without loss of accuracy. However, despite interest from visualization researchers and educators, we have seen little adoption. The main obstacle has not been theoretical soundness but practical setup: the need to install specialized software and host backend services. Without streamlined deployment or user-friendly interfaces, even experts in visualization research find it difficult to incorporate these adaptive instruments into their studies or classrooms.

A similar challenge surfaced in the VILA project. The VILA pipeline successfully used LLMs to generate over a thousand visualization questions and established a structured evaluation rulebook to ensure quality. Yet, its accessibility remains constrained by technical requirements—running Python scripts, managing API keys, and handling output data locally. While the paper has received citations, it has not achieved broad hands-on adoption among educators or researchers who lack substantial programming

experience.

In contrast, the Ripplet system illustrates how lowering technical barriers can improve adoption and impact. By building a web-based, no-installation system and embedding AI assistance directly within teachers' familiar workflows, Ripplet reached dozens of active teachers across more than 10 schools. Its ongoing growth—teachers continuing to use and recommend the tool beyond formal studies—demonstrates how accessible, well-integrated design can bridge the gap between research prototypes and real-world practice. Teachers could log in, immediately begin creating assessments, and iteratively explore AI support without any configuration. This accessibility allowed the research contributions to materialize in daily educational use.

6.3 Mitigating Uneven Risks of AI Reliance Needs

Carefully Designed Support

Across the projects in this dissertation, users and study participants varied widely in their subject areas, experience levels, and familiarity with generative AI tools. These differences can shape not only how they use the systems but also the types of risks they face in doing so. In Ripplet, for example, teachers with more experience often focused on adapting and curating their existing high-quality materials, while less experienced teachers were more likely to rely on AI to generate new content from scratch. Similarly, Teachers with prior AI experience quickly learned to revise prompts to achieve better results, whereas teachers with little AI exposure often struggled at first to understand why the same request could yield different outputs.

These differences suggest that not all users face the same risks with using AI-based

technology for assessment development. Teachers in under-resourced schools—already stretched for time and materials—may be particularly vulnerable to overrelying on AI outputs without the capacity to carefully review them. In contrast, teachers with more experience, training, or institutional support may be better positioned to reuse high-quality content, recognize hallucinations, and adjust their use accordingly. Safeguards and training should be implemented to mitigate these risks. Future systems could embed proactive and explainable support mechanisms that help teachers verify, interpret, and improve AI-generated content as part of their natural workflow. For instance, rather than merely flagging the uncertainty of LLM-generated content, systems could use non-LLM-based verification pipelines to retrieve authoritative references or validated assessment banks to cross-check for correctness. Another possibility is to make AI reasoning more transparent and inspectable. Instead of treating the model’s internal process as a black box, tools could display their reasoning process or confidence explanations that allow teachers to interrogate the rationale behind generated content. For instance, systems could show the inferred learning objective, reasoning steps, or aligned curriculum tags that led to its output to give teachers concrete points for critique or correction. Such features could turn validation from a cognitive burden into an interactive process of sensemaking. For teachers who co-author assessments with colleagues, systems could also explore collaborative verification workflows. Rather than asking each teacher to independently judge AI outputs, they could allow peer co-review or lightweight consensus-building interfaces that pool teacher feedback on the generated assessments. Such distributed review processes could surface common errors and strengthen the collective reliability of AI-generated materials across teachers and schools. Designing for different risks means acknowledging that while AI can be useful for some teachers, it also carries uneven vulnerabilities. Building effective assessment authoring tools therefore requires equipping users—especially those under the most constraints—with support that match their needs.

6.4 Designing Impactful Assessment Systems Requires the Participation of Experts and Educators

Across this dissertation, involving stakeholders in the design and evaluation process has proven essential to building assessment systems that are not only technically sound but also practical, trusted, and impactful.

In VILA, stakeholder involvement was central to ensuring quality. Visualization experts participated in evaluating LLM outputs and collaboratively developed the evaluation rulebook used to guide and standardize quality judgments. Their expertise ensured that the generated questions measured visualization literacy skills. This collaborative evaluation process turned what could have been a purely technical pipeline into a shared framework for defining quality, validity, and relevance in visualization literacy assessment. Similarly, in Ripplet, teachers—the intended end users—played a formative role in shaping the system’s design through an extended codesign process. Working closely with educators revealed their practical and contextual constraints, and these lessons led us to design Ripplet so that it aligns with teachers’ existing workflows and respect teachers’ professional judgment. Together, these cases illustrate that building solutions to support assessment development requires the participation of both domain experts and end users to ensure that solutions are grounded in disciplinary knowledge and connected to the real needs and workflows of educators.

6.5 Bridging Research and Practice Through Integrated Assessment Platforms

A large body of research in education, psychology, and cognitive science offers principles for effective assessment design, such as cognitive models of learning. Yet, these insights rarely make their way into everyday classroom practice. Teachers often do not have the time or resources to interpret academic research or implement complex psychometric recommendations. Future assessment support systems should serve as bridges between research and practice, embedding assessment design principles directly into authoring workflows. Such systems could help teachers not only create assessments more efficiently but also understand why certain test formats or question types better measure student understanding.

Equally important is moving beyond fragmented tools toward integrated platforms that connect assessment authoring, delivery, and grading. Currently, authoring tools and testing platforms often exist as separate systems, requiring teachers to export, reformat, and manually track student results. Building unified ecosystems that connect authoring, delivery, and analytics would allow assessment to become a more holistic and interactive experience. For teachers, this integration could reduce friction and support continuous improvement; for students, it could enable adaptive and personalized feedback. Such platforms could collect and leverage assessment data to help teachers track student progress, recommend next steps, and tailor instruction and future assessments.

Realizing this vision requires attention to both accessibility and participation. As discussed in previous sections, lowering technical barriers remains essential for adoption, and participatory design ensures that systems meet teachers' real needs. Extending

this collaboration to include students can further strengthen the connection between assessment and learning. The next generation of assessment systems should help teachers translate research into practice, unify the processes of development and delivery, and bring teachers and students closer together in the learning cycle. Such systems hold the potential to transform assessment from an isolated task into an ongoing, reflective, and collaborative process—one that continuously improves both teaching and learning.

Chapter 7

Conclusion

This dissertation tackles a persistent challenge in both K-12 education and data visualization research: assessments are vital to teaching and learning, yet remain difficult for educators and researchers to create, adapt, and deliver effectively. Across three projects—adaptive visualization literacy assessments (A-VLAT and A-CALVI), LLM-based item generation through VILA, and the human-LLM collaborative authoring system Ripplet—this work explores how to make assessment development more efficient and scalable while maintaining quality. By integrating AI with human expertise and grounding design in educators’ workflows, we can move toward assessment ecosystems that not only improve efficiency, but also maintain quality and encourage reflection in teaching and learning.

References

- [1] Josh Achiam et al. “GPT-4 Technical Report”. In: *arXiv preprint arXiv:2303.08774* (2023).
- [2] Lenore Adie. “The Development of Teacher Assessment Identity through Participation in Online Moderation”. In: *Assessment in Education: Principles, Policy & Practice* 20.1 (2013), pp. 91–106. DOI: 10.1080/0969594X.2011.650150.
- [3] R. Amar, J. Eagan, and J. Stasko. “Low-level Components of Analytic Activity in Information Visualization”. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. 2005, pp. 111–117. DOI: 10.1109/INFVIS.2005.1532136.
- [4] Kazuhiro Aoyama and Max Stephens. “Graph interpretation aspects of statistical literacy: A Japanese perspective”. In: *Mathematics Education Research Journal* 15.3 (2003), pp. 207–225. DOI: 10.1007/BF03217380.
- [5] American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. *Standards for Educational and Psychological Testing*. American Educational Research Association, 2014.
- [6] Frank B Baker. *The Basics of Item Response Theory*. ERIC, 2001.
- [7] Nicole Barnes, Helenrose Fives, and Charity M. Dacey. “Teachers’ beliefs about assessment”. English. In: *International Handbook of Research on Teachers’ Beliefs*. Taylor and Francis, 2014, pp. 284–300. ISBN: 9780415539227. DOI: 10.4324/9780203108437-25.
- [8] Nicole Barnes, Helenrose Fives, and Charity M. Dacey. “U.S. teachers’ conceptions of the purposes of assessment”. In: *Teaching and Teacher Education* 65 (2017), pp. 107–116. ISSN: 0742-051X. DOI: 10.1016/j.tate.2017.02.017.

- [9] David Beiser, Milkie Vu, and Robert Gibbons. “Test-Retest Reliability of a Computerized Adaptive Depression Screener”. In: *Psychiatric Services* 67.9 (2016). PMID: 27079989, pp. 1039–1041. DOI: 10.1176/appi.ps.201500304. eprint: <https://doi.org/10.1176/appi.ps.201500304>. URL: <https://doi.org/10.1176/appi.ps.201500304>.
- [10] Randy Elliot Bennett and Drew H. Gitomer. “Transforming K–12 Assessment: Integrating Accountability Testing, Formative Assessment and Professional Support”. In: (2009), pp. 43–61. DOI: 10.1007/978-1-4020-9964-9_3.
- [11] Paul Black and Dylan Wiliam. “Developing the theory of formative assessment”. In: *Educational Assessment, Evaluation and Accountability (formerly: Journal of Personnel Evaluation in Education)* 21.1 (2009). 10.1007/s11092-008-9068-5, pp. 5–31. ISSN: 1874-8600.
- [12] Paul Black and Dylan Wiliam. *Inside the black box: Raising standards through classroom assessment*. Granada Learning, 1998.
- [13] College Board. *Advance Placement (AP) Program*. <https://ap.collegeboard.org/>. 2025.
- [14] Katy Börner, Andreas Bueckle, and Michael Ginda. “Data Visualization Literacy: Definitions, Conceptual Frameworks, Exercises, and Assessments”. In: *Proceedings of the National Academy of Sciences* 116.6 (2019), pp. 1857–1864. DOI: 10.1073/pnas.1807180116.
- [15] Jeremy Boy et al. “A Principled Way of Assessing Visualization Literacy”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1963–1972. DOI: 10.1109/TVCG.2014.2346984.
- [16] Virginia Braun and Victoria Clarke. “Using thematic analysis in psychology”. In: *Qualitative Research in Psychology* 3.2 (2006), pp. 77–101. DOI: 10.1191/1478088706qp063oa.
- [17] Susan M. Brookhart. “Teaching About Communicating Assessment Results and Grading”. In: *Educational Measurement: Issues and Practice* 18.1 (1999), pp. 5–13.
- [18] Gavin T. L. Brown. “Teachers’ conceptions of assessment: implications for policy and professional development”. In: *Assessment in Education: Principles, Policy & Practice* 11.3 (2004), pp. 301–318.

- [19] Gavin T. L. Brown. “Teachers’ Conceptions of Assessment: Validation of an Abridged Version”. In: *Psychological Reports* 99.1 (2006), pp. 166–170. DOI: 10.2466/pr0.99.1.166–170.
- [20] Sally Brown. “Assessment for learning”. In: *Learning and teaching in higher education* 1 (2005), pp. 81–89.
- [21] Tom B. Brown et al. “Language Models Are Few-Shot Learners”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS’20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546. DOI: 10.48550/arXiv.2005.14165.
- [22] A. Burns et al. “How to evaluate data visualizations across different levels of understanding”. In: *2020 IEEE Workshop on Evaluation and Beyond - Methodological Approaches to Visualization (BELIV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2020, pp. 19–28. DOI: 10.1109/BELIV51497.2020.00010. URL: <https://doi.ieeecomputersociety.org/10.1109/BELIV51497.2020.00010>.
- [23] “Capturing Greater Context for Question Generation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (Apr. 2020), pp. 9065–9072. DOI: 10.1609/aaai.v34i05.6440.
- [24] James Carifio and Rocco Perla. “Resolving the 50-year debate around using and misusing Likert scales.” eng. In: *Med Educ* 42.12 (Dec. 2008), pp. 1150–1152. ISSN: 1365-2923 (Electronic); 0308-0110 (Linking). DOI: 10.1111/j.1365-2923.2008.03172.x.
- [25] Kevin D Carlson and Andrew O Herdman. “Understanding the Impact of Convergent Validity on Research Results”. In: *Organizational Research Methods* 15.1 (2012), pp. 17–32. DOI: 10.1177/1094428110392383.
- [26] Guanliang Chen et al. “LearningQ: A Large-Scale Dataset for Educational Question Generation”. In: *Proceedings of the International AAAI Conference on Web and Social Media* 12.1 (2018). DOI: 10.1609/icwsm.v12i1.14987.
- [27] Kun-Ting Chen et al. “Rotate or Wrap? Interactive Visualisations of Cyclical Data on Cylindrical or Toroidal Topologies”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 727–736. DOI: 10.1109/TVCG.2021.3114693.
- [28] Mark Chen et al. “Evaluating Large Language Models Trained on Code”. In: *arXiv preprint arXiv:2107.03374* (2021).

- [29] Z. Chen et al. “Beyond Generating Code: Evaluating GPT on a Data Visualization Course”. In: *2023 IEEE VIS Workshop on Visualization Education, Literacy, and Activities (EduVis)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2023, pp. 16–21. DOI: 10.1109/EduVis60792.2023.00009. URL: <https://doi.ieeecomputersociety.org/10.1109/EduVis60792.2023.00009>.
- [30] Yi Cheng et al. “Guiding the Growth: Difficulty-Controllable Question Generation through Step-by-Step Rewriting”. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*. 2021, pp. 5968–5978. DOI: 10.18653/v1/2021.acl-long.465.
- [31] Zirui Cheng, Jingfei Xu, and Haojian Jin. “TreeQuestion: Assessing Conceptual Learning Outcomes with LLM-Generated Multiple-Choice Questions”. In: *Proceedings of the ACM on Human-Computer Interaction* 8 (Nov. 2024). DOI: 10.1145/3686970.
- [32] Erin Cherry and Celine Latulipe. “Quantifying the Creativity Support of Digital Tools through the Creativity Support Index”. In: *ACM Trans. Comput.-Hum. Interact.* 21.4 (June 2014). ISSN: 1073-0516. DOI: 10.1145/2617588. URL: <https://doi.org/10.1145/2617588>.
- [33] Ching-Lan Chin and Grace Yao. “Convergent Validity”. In: *Encyclopedia of Quality of Life and Well-Being Research*. Dordrecht: Springer, 2014, pp. 1275–1276. ISBN: 978-94-007-0753-5. DOI: 10.1007/978-94-007-0753-5_573.
- [34] Domenic V Cicchetti. “Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology.” In: *Psychological Assessment* 6.4 (1994), pp. 284–290. DOI: 10.1037/1040-3590.6.4.284.
- [35] Ronald Jay Cohen, W. Joel Schneider, and Renée Tobin. *Psychological Testing and Assessment*. 10th. New York, NY: McGraw Hill LLC, 2022. ISBN: 1260837025.
- [36] CollegeBoard. *Transitioning to a Digital SAT*. <https://professionals.collegeboard.org/pdf/digital-sat-faculty-guide-v3-ada-v0.pdf>. 2022.
- [37] Andrew Coombs et al. “Changing approaches to classroom assessment: An empirical study across teacher career stages”. In: *Teaching and Teacher Education* 71 (2018), pp. 134–144. DOI: 10.1016/j.tate.2017.12.010.
- [38] Yuan Cui et al. “Adaptive Assessment of Visualization Literacy”. In: *IEEE Transactions on Visualization and Computer Graphics* 30.1 (2024), pp. 628–637. DOI: 10.1109/TVCG.2023.3327165.

- [39] Yuan Cui et al. “Promises and Pitfalls: Using Large Language Models to Generate Visualization Items”. In: *IEEE Transactions on Visualization and Computer Graphics* 31.1 (2025), pp. 1094–1104. DOI: 10.1109/TVCG.2024.3456309.
- [40] ROBERT DelMas, Joan Garfield, and Ann Ooms. “Using Assessment Items to Study Students’ Difficulty Reading and Interpreting Graphical Representations of distributions”. In: *Fourth Forum on Statistical Reasoning, Thinking, and Literacy (SRTL-4)*. Auckland, New Zealand, 2005.
- [41] Christopher DeLuca and Don A. Klinger. “Assessment literacy development: identifying gaps in teacher candidates’ learning”. In: *Assessment in Education: Principles, Policy & Practice* 17.4 (2010), pp. 419–438. ISSN: 0969-594X. DOI: 10.1080/0969594x.2010.516643.
- [42] Christopher DeLuca, Danielle LaPointe-McEwan, and Ulemu Luhanga. “Teacher assessment literacy: A review of international standards and measures”. In: *Educational Assessment, Evaluation and Accountability* 28.3 (2016), pp. 251–272. ISSN: 1874-8597. DOI: 10.1007/s11092-015-9233-6.
- [43] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Tamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://aclanthology.org/N19-1423>.
- [44] Y. Ding et al. “reVISit: Supporting Scalable Evaluation of Interactive Visualizations”. In: *2023 IEEE Visualization and Visual Analytics (VIS)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2023, pp. 31–35. DOI: 10.1109/VIS54172.2023.00015. URL: <https://doi.ieeecomputersociety.org/10.1109/VIS54172.2023.00015>.
- [45] Steven M. Downing. “The Effects of Violating Standard Item Writing Principles on Tests and Students: The Consequences of Using Flawed Test Items on Achievement Examinations in Medical Education”. In: *Advances in Health Sciences Education* 10.2 (2005), pp. 133–143. ISSN: 1382-4996. DOI: 10.1007/s10459-004-4019-5.
- [46] Sabina Elkins et al. “How Useful Are Educational Questions Generated by Large Language Models?” In: *Communications in Computer and Information Science* (2023), pp. 536–542. DOI: 10.1007/978-3-031-36336-8_83.

- [47] ETS. *GRE General Test Structure*. <https://www.ets.org/gre/test-takers/general-test/prepare/test-structure.html>. 2023.
- [48] Haoxiang Fan et al. “LessonPlanner: Assisting Novice Teachers to Prepare Pedagogy-Driven Lesson Plans with Large Language Models”. In: 2024. DOI: 10.1145/3654777.3676390.
- [49] Suzanne Fergus, Michelle Botha, and Mehrnoosh Ostovar. “Evaluating Academic Answers Generated Using ChatGPT”. In: *Journal of Chemical Education* 100.4 (Apr. 2023), pp. 1672–1675.
- [50] Sandra Ferketich. “Focus on psychometrics. Aspects of item analysis”. In: *Research in Nursing & Health* 14.2 (1991), pp. 165–168. DOI: 10.1002/nur.4770140211.
- [51] Barbara Foorman et al. “Using Computer-Adaptive Assessments of Literacy to Monitor the Progress of English Learner Students”. In: *Regional Educational Laboratory Southeast (REL 2016-149)* (2016).
- [52] Jonas Frich et al. “Mapping the Landscape of Creativity Support Tools in HCI”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–18. ISBN: 9781450359702. DOI: 10.1145/3290605.3300619. URL: <https://doi.org/10.1145/3290605.3300619>.
- [53] Yifan Gao et al. “Difficulty Controllable Generation of Reading Comprehension Questions”. In: *Proceedings of the International Joint Conference on Artificial Intelligence* (2018). DOI: 10.48550/arxiv.1807.03586.
- [54] John Gardner. “Assessment for learning: A compelling conceptualization”. In: *Assessment and learning* (2006), pp. 197–204.
- [55] Lily W. Ge, Yuan Cui, and Matthew Kay. “CALVI: Critical Thinking Assessment for Literacy in Visualizations”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI’23. Hamburg, Germany: Association for Computing Machinery, 2023, pp. 1–18. DOI: 10.1145/3544548.3581406.
- [56] Abdallah Ghaicha. “Theoretical Framework for Educational Assessment: A Synoptic Review.” In: *Journal of Education and Practice* 7.24 (2016), pp. 212–231.
- [57] Fernanda Belén Ghio et al. “Preliminary Development of an Item Bank and an Adaptive Test in Mathematical Knowledge for University Students”. In: *European*

- Journal of Science and Mathematics Education* 10.3 (2022), pp. 352–365. DOI: 10 . 30935/scimath/11968.
- [58] Chris Gibbons et al. “Electronic Quality of Life Assessment Using Computer-Adaptive Testing”. In: *J Med Internet Res* 18.9 (2016), e240. DOI: 10.2196/jmir.6053.
 - [59] Robert D Gibbons and Frank V deGruy. “Without Wasting a Word: Extreme Improvements in Efficiency and Accuracy Using Computerized Adaptive Testing for Mental Health Disorders (CAT-MH)”. In: *Current Psychiatry Reports* 21.67 (2019). DOI: 10.1007/s11920-019-1053-9.
 - [60] Robert D Gibbons et al. “Development of the CAT-ANX: A Computerized Adaptive Test for Anxiety”. In: *American Journal of Psychiatry* 171.2 (2014), pp. 187–194. DOI: 10.1176/appi.ajp.2013.13020178.
 - [61] Mark J Gierl and Thomas M Haladyna. *Automatic Item Generation: Theory and Practice*. Routledge, 2012. DOI: <https://doi.org/10.4324/9780203803912>.
 - [62] Bert F Green et al. “Technical Guidelines for Assessing Computerized Adaptive Tests”. In: *Journal of Educational measurement* 21.4 (1984), pp. 347–360. DOI: 10 . 1111/j.1745-3984.1984.tb01039.x.
 - [63] Daniel Guinart et al. “Development and Validation of a Computerized Adaptive Assessment Tool for Discrimination and Measurement of Psychotic Symptoms”. In: *Schizophrenia Bulletin* 47.3 (2021), pp. 644–652. DOI: 10.1093/schbul/sbaa168.
 - [64] Thomas M Haladyna and Michael C Rodriguez. *Developing and Validating Test Items*. Routledge, 2013. DOI: 10.4324/9780203850381.
 - [65] Ching Nam Hang, Chee Wei Tan, and Pei-Duo Yu. “MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning”. In: *IEEE Access* 12 (2024), pp. 102261–102273. DOI: 10.1109/ACCESS.2024.3420709.
 - [66] Forrest Huang et al. “Automatic Macro Mining from Interaction Traces at Scale”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: 10 . 1145 / 3613904 . 3642074. URL: <https://doi.org/10.1145/3613904.3642074>.
 - [67] Elaine Huynh et al. “Designing Narrative-Focused Role-Playing Games for Visualization Literacy in Young Children”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 924–934. DOI: 10.1109/TVCG.2020.3030464.

- [68] Ziwei Ji et al. “Survey of Hallucination in Natural Language Generation”. In: 55.12 (Mar. 2023). ISSN: 0360-0300. DOI: 10.1145/3571730. URL: <https://doi.org/10.1145/3571730>.
- [69] Alark Joshi et al. “Evaluating the Recommendations of LLMs to Teach a Visualization Technique Using Bloom’s Taxonomy”. In: *Electronic Imaging* 36.1 (2024), pp. 360-1–360-1. DOI: 10.2352/EI.2024.36.1.VDA-360. URL: <https://library.imaging.org/ei/articles/36/1/VDA-360>.
- [70] Kushal Kafle et al. “DVQA: Understanding Data Visualizations via Question Answering”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5648–5656. DOI: 10.1109/CVPR.2018.00592.
- [71] Adam Tauman Kalai and Santosh S Vempala. “Calibrated Language Models Must Hallucinate”. In: *arXiv preprint arXiv:2311.14648* (2023).
- [72] Sasikiran Kandula et al. “A New Adaptive Testing Algorithm for Shortening Health Literacy Assessments”. In: *BMC Medical Informatics and Decision Making* 11.52 (2011). DOI: 10.1186/1472-6947-11-52.
- [73] Wenhui Kang et al. “TutorCraftEase: Enhancing Pedagogical Question Creation with Large Language Models”. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2025. DOI: 10.1145/3706598.3713731.
- [74] KaTeX Contributors. *KaTeX*. <https://katex.org/>. 2013.
- [75] Gordon Kindlmann and Carlos Scheidegger. “An Algebraic Process for Visualization Design”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2181–2190. DOI: 10.1109/TVCG.2014.2346325.
- [76] David R Krathwohl. “A Revision of Bloom’s Taxonomy: An Overview”. In: *Theory into practice* 41.4 (2002), pp. 212–218. DOI: 10.1207/s15430421tip4104_2.
- [77] Stefan Küchemann et al. “Can ChatGPT support prospective teachers in physics task development?” In: *Physical Review Physics Education Research* 19 (2 2023), p. 020128. DOI: 10.1103/PhysRevPhysEducRes.19.020128.
- [78] Ghader Kurdi et al. “A Systematic Review of Automatic Question Generation for Educational Purposes”. In: *International Journal of Artificial Intelligence in Education* 30.1 (2020), pp. 121–204. ISSN: 1560-4292. DOI: 10.1007/s40593-019-00186-y.

- [79] Suzanne Lane, Mark R Raymond, Thomas M Haladyna, et al. *Handbook of Test Development*. Vol. 2. Routledge New York, NY, 2016. DOI: <https://doi.org/10.4324/9780203102961>.
- [80] Hui Min Lee et al. “Shorter and Simpler: Customizing Generative AI Responses Can Increase Satisfaction, Credibility, and Fact-Checking Intentions”. In: *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '25. Association for Computing Machinery, 2025. ISBN: 9798400713958. DOI: 10.1145/3706599.3719749. URL: <https://doi.org/10.1145/3706599.3719749>.
- [81] Sukwon Lee, Sung-Hee Kim, and Bum Chul Kwon. “VLAT: Development of a Visualization Literacy Assessment Test”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 551–560. DOI: 10.1109/TVCG.2016.2598920.
- [82] Elsie Lee-Robbins, Shiqing He, and Eytan Adar. “Learning Objectives, Insights, and Assessments: How Specification Formats Impact Design”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 676–685. DOI: 10.1109/TVCG.2021.3114811.
- [83] David Liljequist, Britt Elfving, and Kirsti Skavberg Roaldsen. “Intraclass correlation –A discussion and demonstration of basic features”. In: *PLoS ONE* 14.7 (2019), e0219854. DOI: 10.1371/journal.pone.0219854.
- [84] Anne Looney et al. “Reconceptualising the role of teachers as assessors: teacher assessment identity”. In: *Assessment in Education: Principles, Policy & Practice* 25.5 (2018), pp. 442–467. DOI: 10.1080/0969594X.2016.1268090.
- [85] Xinyi Lu et al. “ReadingQuizMaker: A Human-NLP Collaborative System that Supports Instructors to Design High-Quality Reading Quiz Questions”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394215. DOI: 10.1145/3544548.3580957.
- [86] Sonja Lutovac and Maria Assunção Flores. “Conceptions of assessment in pre-service teachers’ narratives of students’ failure”. In: *Cambridge Journal of Education* 52.1 (2022), pp. 55–71.
- [87] Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. “A Novel Multi-Stage Prompting Approach for Language Agnostic MCQ Generation Using GPT”. In: *Advances in Information Retrieval*. 2024, pp. 268–277. ISBN: 978-3-031-56063-7.

- [88] Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. “Exploring the capabilities of prompted large language models in educational and assessment applications”. In: *arXiv preprint arXiv:2405.11579* (2024).
- [89] Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. “Harnessing the Power of Prompt-based Techniques for Generating School-Level Questions using Large Language Models”. In: *Proceedings of the Annual Meeting of the Forum for Information Retrieval Evaluation*. 2024, pp. 30–39. DOI: 10.1145/3632754.3632755.
- [90] Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. “How Effective is GPT-4 Turbo in Generating School-Level Questions from Textbooks Based on Bloom’s Revised Taxonomy?” In: *arXiv* (2024). DOI: 10.48550/arxiv.2406.15211.
- [91] Ahmed Masry et al. “ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning”. In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2263–2279. DOI: 10.18653/v1/2022.findings-acl.177. URL: <https://aclanthology.org/2022.findings-acl.177>.
- [92] Damien Masson et al. “DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI ’24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: 10.1145/3613904.3642462.
- [93] Granville J Matheson. “We Need to Talk about Reliability: Making Better Use of Test-retest Studies for Study Design and Interpretation”. In: *PeerJ* 7 (2019), e6918. DOI: 10.7717/peerj.6918.
- [94] James H. McMillan. “Understanding and Improving Teachers’ Classroom Assessment Decision Making: Implications for Theory and Practice”. In: *Educational Measurement: Issues and Practice* 22.4 (2003). 10.1111/j.1745-3992.2003.tb00142.x, pp. 34–43.
- [95] Craig A. Mertler. “Teachers’ assessment knowledge and their perceptions of the impact of classroom assessment professional development”. In: *Improving Schools* 12.2 (2009), pp. 101–113.
- [96] N. Methani et al. “PlotQA: Reasoning over Scientific Plots”. In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2020, pp. 1516–1525. DOI: 10.1109/WACV45572.2020.

9093523. URL: <https://doi.ieeecomputersociety.org/10.1109/WACV45572.2020.9093523>.
- [97] Ning Miao, Yee Whye Teh, and Tom Rainforth. “SelfCheck: Using LLMs to Zero-Shot Check Their Own Step-by-Step Reasoning”. In: *arXiv preprint arXiv:2308.00436* (2023).
 - [98] Steven Moore et al. “Assessing the Quality of Student-Generated Short Answer Questions Using GPT-3”. In: *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption*. 2022, pp. 243–257.
 - [99] Nikahat Mulla and Prachi Gharpure. “Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications”. In: *Progress in Artificial Intelligence* 12.1 (2023), pp. 1–32. ISSN: 2192-6352. DOI: 10.1007/s13748-023-00295-9.
 - [100] Geoff Norman. “Likert scales, levels of measurement and the ”laws” of statistics.” eng. In: *Adv Health Sci Educ Theory Pract* 15.5 (Dec. 2010), pp. 625–632. ISSN: 1573-1677 (Electronic); 1382-4996 (Linking). DOI: 10.1007/s10459-010-9222-y.
 - [101] OpenAI. *Documentation: How Should I Set The Temperature Parameter*. <https://platform.openai.com/docs/guides/text-generation/how-should-i-set-the-temperature-parameter>.
 - [102] OpenAI. *Documentation: Prompt Engineering*. <https://platform.openai.com/docs/guides/prompt-engineering>.
 - [103] OpenAI. *Documentation: Reproducible Outputs*. <https://platform.openai.com/docs/guides/text-generation/reproducible-outputs>.
 - [104] *Our World in Data*. <https://ourworldindata.org/>. 2024.
 - [105] Liangming Pan et al. “Semantic Graphs for Generating Deep Questions”. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 1463–1475. DOI: 10.18653/v1/2020.acl-main.135.
 - [106] Serafina Pastore and Heidi L. Andrade. “Teacher assessment literacy: A three-dimensional model”. In: *Teaching and Teacher Education* 84 (2019), pp. 128–138.
 - [107] James W. Pellegrino, Naomi Chudowsky, and Robert Glaser, eds. *Knowing What Students Know: The Science and Design of Educational Assessment*. Washington, DC:

- The National Academies Press, 2001. ISBN: 978-0-309-29322-8. DOI: 10 . 17226 / 10019.
- [108] Karen Peterman et al. “Measuring Primary Students’ Graph Interpretation Skills Via a Performance Assessment: A Case Study in Instrument Development”. In: *International Journal of Science Education* 37.17 (2015), pp. 2787–2808. DOI: 10 . 1080 / 09500693 . 2015 . 1105399.
 - [109] Denise F Polit. “Getting Serious About Test–retest Reliability: A Critique of Retest Research and Some Recommendations”. In: *Quality of Life Research* 23 (2014), pp. 1713–1720. DOI: 10 . 1007 / s11136 - 014 - 0632 - 9.
 - [110] Denise F. Polit, Cheryl Tatano Beck, and Steven V. Owen. “Is the CVI an acceptable indicator of content validity? Appraisal and recommendations”. In: *Research in Nursing & Health* 30.4 (2007), pp. 459–467. DOI: <https://doi.org/10.1002/nur.20199>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nur.20199>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nur.20199>.
 - [111] Ofir Press et al. “Measuring and Narrowing the Compositionality Gap in Language Models”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5687–5711. DOI: 10 . 18653 / v1 / 2023 . findings - emnlp . 378. URL: <https://aclanthology.org/2023.findings-emnlp.378>.
 - [112] PrismJS Contributors. *PrismJS*. <https://prismjs.com/>. 2012.
 - [113] Vatsal Raina and Mark Gales. “Multiple-choice question generation: Towards an automated assessment framework”. In: *arXiv preprint arXiv:2209.11830* (2022).
 - [114] React Syntax Highlighter Contributors. *React Syntax Highlighter*. <https://github.com/react-syntax-highlighter/react-syntax-highlighter>. 2016.
 - [115] rehype Contributors. *rehype-katex*. <https://www.npmjs.com/package/rehype-katex>. 2017.
 - [116] remarkjs Contributors. *remark-math*. <https://github.com/remarkjs/remark-math>. 2016.
 - [117] Ana Remesal. “Primary and secondary teachers’ conceptions of assessment: A qualitative study”. In: *Teaching and Teacher Education* 27.2 (2011). 10.1016/j.tate.2010.09.017, pp. 472–482. ISSN: 0742-051X.

- [118] Laria Reynolds and Kyle McDonell. “Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm”. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI EA '21. , Yokohama, Japan, Association for Computing Machinery, 2021. ISBN: 9781450380959. DOI: 10.1145/3411763.3451760. URL: <https://doi.org/10.1145/3411763.3451760>.
- [119] Mohi Reza et al. “ABSScribe: Rapid Exploration & Organization of Multiple Writing Variations in Human-AI Co-Writing Tasks using Large Language Models”. In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: 10.1145/3613904.3641899.
- [120] Mohi Reza et al. “PromptHive: Bringing Subject Matter Experts Back to the Forefront with Collaborative Prompt Engineering for Educational Content Creation”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3714051. URL: <https://doi.org/10.1145/3706598.3714051>.
- [121] Marzieh Rezaie and Mohammad Golshan. “Computer Adaptive Test (CAT): Advantages and Limitations”. In: *International Journal of Educational Investigations* 2.5 (2015), pp. 128–137.
- [122] Lawrence M Rudner. “Implementing the Graduate Management Admission Test Computerized Adaptive Test”. In: *Elements of Adaptive Testing, Statistics for Social and Behavioral Sciences*. New York, NY: Springer, 2009, pp. 151–165. DOI: 10.1007/978-0-387-85461-8_8.
- [123] Daniel O Segall. “Computerized Adaptive Testing”. In: *Encyclopedia of social measurement* 1 (2005), pp. 429–438.
- [124] Richard J Stiggins. “Revitalizing classroom assessment: The highest instructional priority”. In: *The Phi Delta Kappan* 69.5 (1988). <https://www.jstor.org/stable/20403636>, pp. 363–368.
- [125] Richard J. Stiggins. “The Unfulfilled Promise of Classroom Assessment”. In: *Educational Measurement: Issues and Practice* 20.3 (2001), pp. 5–15.
- [126] Richard J. Stiggins, Nancy F. Conklin, and Nancy J. Bridgeford. “Classroom assessment: A key to effective education”. In: *Educational Measurement: Issues and Practice* 5.2 (1986). 10.1111/j.1745-3992.1986.tb00473.x, pp. 5–17.

- [127] John M. Stonehouse and Guy J. Forrester. “Robustness of the t and U tests under combined assumption violations”. In: *Journal of Applied Statistics* 25.1 (1998), pp. 63–74. DOI: 10.1080/02664769823304. eprint: <https://doi.org/10.1080/02664769823304>. URL: <https://doi.org/10.1080/02664769823304>.
- [128] Zachari Swiecki et al. “Assessment in the age of artificial intelligence”. In: *Computers and Education: Artificial Intelligence* 3 (2022), p. 100075. ISSN: 2666-920X. DOI: 10.1016/j.caeai.2022.100075.
- [129] Sara Tandon, Alfie Abdul-Rahman, and Rita Borgo. “Measuring Effects of Spatial Visualization and Domain on Visualization Task Performance: A Comparative Study”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.1 (2023), pp. 668–678. DOI: 10.1109/TVCG.2022.3209491.
- [130] Maddalena Taras. “Summative assessment: The missing link for formative assessment”. In: *Journal of Further and Higher Education* 33.1 (2009). 10.1080/03098770802638671, pp. 57–69.
- [131] Marie Tarrant et al. “The frequency of item writing flaws in multiple-choice questions used in high stakes nursing assessments”. In: *Nurse Education Today* 26.8 (2006), pp. 662–671. ISSN: 0260-6917. DOI: 10.1016/j.nedt.2006.07.006.
- [132] Nathan A Thompson and David A Weiss. “A Framework for the Development of Computerized Adaptive Tests”. In: *Practical Assessment, Research, and Evaluation* 16.1 (2011). DOI: 10.7275/wqzt-9427.
- [133] Robin D Tierney. “Fairness in classroom assessment”. In: *SAGE handbook of research on classroom assessment* (2012), pp. 125–145.
- [134] Luu Anh Tuan, Darsh Shah, and Regina Barzilay. “Capturing Greater Context for Question Generation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (2020), pp. 9065–9072. DOI: 10.1609/aaai.v34i05.6440.
- [135] Masaki Uto, Yuto Tomikawa, and Ayaka Suzuki. “Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory”. In: *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*. 2023, pp. 119–129. DOI: 10.18653/v1/2023.bea-1.10.
- [136] Veronica Villarroel et al. “Authentic Assessment: Creating A Blueprint for Course Design”. In: *Assessment & Evaluation in Higher Education* 43.5 (2018), pp. 840–854. DOI: 10.1080/02602938.2017.1412396.

- [137] Howard Wainer et al. *Computerized Adaptive Testing: A Primer*. 2nd. New York: Routledge, 2000. DOI: 10.4324/9781410605931.
- [138] Otto B Walter et al. “Development and evaluation of a computer adaptive test for ‘Anxiety’ (Anxiety-CAT)”. In: *Quality of Life Research* 16 (2007), pp. 143–155. DOI: 10.1007/s11136-007-9191-7.
- [139] Dakuo Wang et al. “From Human-Human Collaboration to Human-AI Collaboration: Designing AI Systems That Can Work Together with People”. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI EA ’20. <conf-loc>, <city>Honolulu</city>, <state>HI</state>, <country>USA</country>, </conf-loc>: Association for Computing Machinery, 2020, pp. 1–6. ISBN: 9781450368193. DOI: 10.1145/3334480.3381069. URL: <https://doi.org/10.1145/3334480.3381069>.
- [140] Xu Wang et al. “Towards Process-Oriented, Modular, and Versatile Question Generation that Meets Educational Needs”. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 10.18653/v1/2022.naacl-main.22. Association for Computational Linguistics, 2022, pp. 291–302.
- [141] Zichao Wang et al. “QG-net: a data-driven question generation model for educational content”. In: *Proceedings of the Annual ACM Conference on Learning at Scale*. 2018. DOI: 10.1145/3231644.3231654.
- [142] Jason Wei et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24824–24837. DOI: <https://doi.org/10.48550/arXiv.2201.11903>. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.
- [143] WestEd. *The Next Generation Science Standards*. <https://www.nextgenscience.org/>. 2025.
- [144] Nico Willert and Jonathan Thiemann. “Template-Based Generator for Single-Choice Questions”. In: *Technology, Knowledge and Learning* 29.1 (2024), pp. 355–370. DOI: <https://doi.org/10.1007/s10758-023-09659-5>.
- [145] Jill Willis. In: *Assessment Matters* 2 (2010), pp. 65–84. DOI: [doi/10.3316/informit.330525572573096](https://doi.org/10.3316/informit.330525572573096).

- [146] Rafael Wilms, Ralf Lanwehr, and Andreas Kastenmüller. “Do We Overestimate the Within-Variability? The Impact of Measurement Error on Intraclass Coefficient Estimation”. In: *Frontiers in Psychology* 11 (2020), p. 825. DOI: 10.3389/fpsyg.2020.00825.
- [147] Chris Woolston. “Psychology journal bans P values”. In: *Nature* 519.7541 (2015), pp. 9–9. DOI: 10.1038/519009f. URL: <https://doi.org/10.1038/519009f>.
- [148] Yuxi Xie et al. “Exploring Question-Specific Rewards for Generating Deep Questions”. In: *Proceedings of the International Conference on Computational Linguistics*. 2020, pp. 2534–2546. DOI: 10.18653/v1/2020.coling-main.228.
- [149] Yueting Xu and Gavin T.L. Brown. “Teacher assessment literacy in practice: A reconceptualization”. In: *Teaching and Teacher Education* 58 (2016), pp. 149–162. DOI: 10.1016/j.tate.2016.05.010.
- [150] Ryan Yen et al. “CoLadder: Manipulating Code Generation via Multi-Level Blocks”. In: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. UIST ’24. Pittsburgh, PA, USA: Association for Computing Machinery, 2024. ISBN: 9798400706288. DOI: 10.1145/3654777.3676357. URL: <https://doi.org/10.1145/3654777.3676357>.
- [151] Guanhua Zhang et al. “SummAct: Uncovering User Intentions Through Interactive Behaviour Summarisation”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI ’25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713190. URL: <https://doi.org/10.1145/3706598.3713190>.