

AmlWrite: Exploring Scalable One-on-One Handwriting-Based Tutoring for Mathematical Problem-Solving with an LLM-Powered AI Tutor

Ziyi Liu*
Purdue University
West Lafayette, IN, USA
liu1362@purdue.edu

Yuzhao Chen*
Purdue University
West Lafayette, IN, USA
chen4863@purdue.edu

Haoyu Ji
purdue university
West Lafayette, IN, USA
ji223@purdue.edu

Runlin Duan
Mechanical Engineering,C
Design Lab
Purdue University
West Lafayette, IN, USA
duan92@purdue.edu

Zhengzhe Zhu
Purdue University
West Lafayette, IN, USA
robertzhu1994@gmail.com

Xiyun Hu
School of Mechanical
Engineering
Purdue University
West Lafayette, IN, USA
hu690@purdue.edu

Kylie Pepler
Informatics and Education
University of California -
Irvine
Irvine, CA, USA
kpepler@iu.edu

Karthik Ramani
School of Mechanical
Engineering
Purdue University
West Lafayette, IN, USA
ramani@purdue.edu

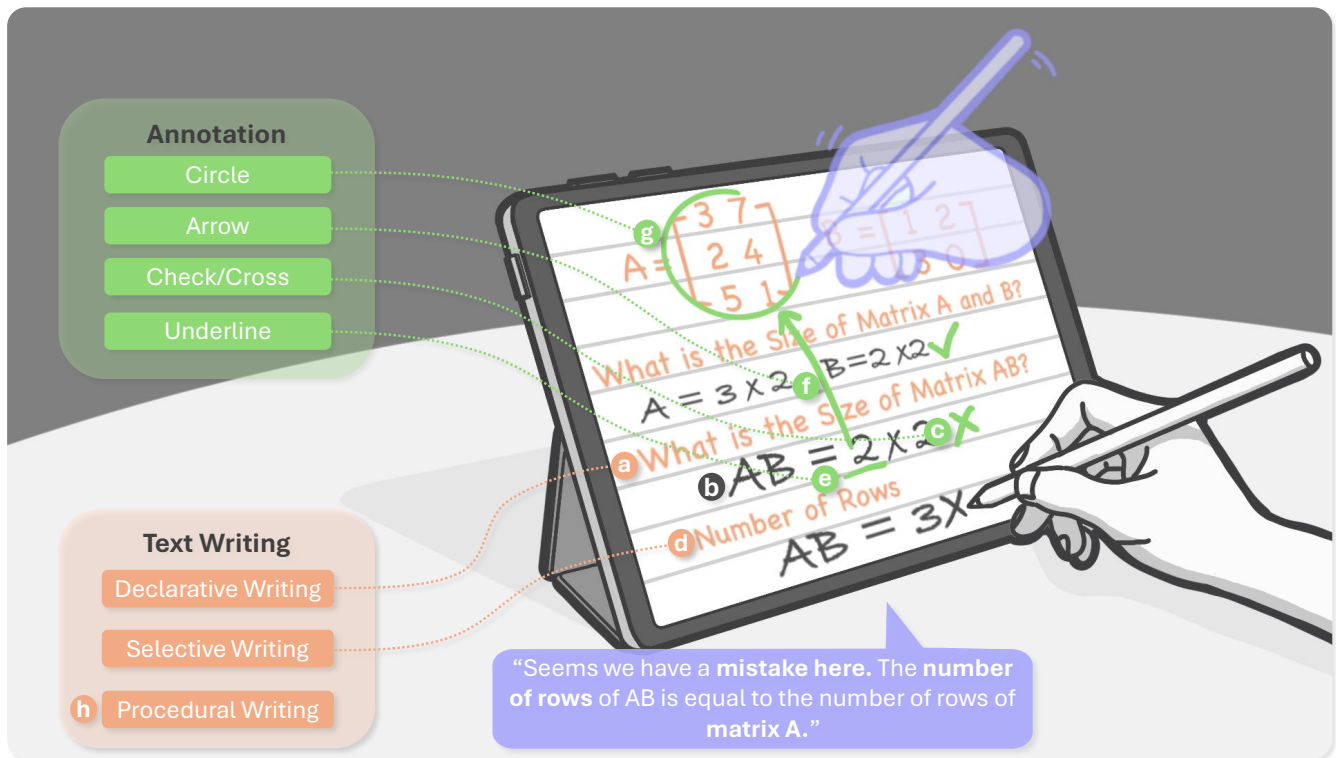


Figure 1: AmlWrite is an LLM-powered AI tutoring system that supports mathematical problem-solving through real-time handwriting interaction. In a matrix multiplication lesson, (a) the AI tutor asks, “What is the size of matrix AB?” (b) When the student writes an incorrect answer, the tutor provides synchronized verbal and handwritten feedback. (c) A cross mark appears at the error when the tutor says “mistake here.” (e) The relevant number is underlined when mentioning (d) the number of rows of AB. (f) An arrow links the underlined number to matrix A, (g) which is circled to illustrate the relationship. The student then corrects the answer. (h) If difficulties persist, the tutor provides a procedural demonstration of the current step.

Abstract

Real-time handwriting interactions between tutors and students—where tutors observe individual problem-solving processes, provide personalized annotations, and adapt explanations based on students’ work—are fundamental to effective STEM tutoring. However, scaling such personalized handwriting-based tutoring remains challenging—human tutors cannot be available to every student on demand, and current online platforms often fail to recreate equivalent learning experiences. As an initial step toward tackling this challenge, we present AmlWrite, an LLM-powered AI tutoring system for mathematical problem-solving that provides real-time co-speech handwriting interactions on tablet devices, instantiated here as a case study in linear algebra. We conducted a within-subjects study ($N = 40$) comparing AmlWrite to a text-based AI tutor on two linear algebra topics. Our case study demonstrates how a multimodal AI tutor can preserve the pedagogical benefits of handwriting-based math tutoring and offer a potential path toward more scalable one-on-one STEM tutoring.

CCS Concepts

• **Human-centered computing** → **Natural language interfaces; Collaborative interaction.**

Keywords

pedagogical agent, collaborative learning, large language model

ACM Reference Format:

Ziyi Liu, Yuzhao Chen, Haoyu Ji, Runlin Duan, Zhengzhe Zhu, Xiyun Hu, Kylie Pepler, and Karthik Ramani. 2026. AmlWrite: Exploring Scalable One-on-One Handwriting-Based Tutoring for Mathematical Problem-Solving with an LLM-Powered AI Tutor. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*, April 13–17, 2026, Barcelona, Spain. ACM, New York, NY, USA, 28 pages. <https://doi.org/10.1145/3772318.3790935>

1 Introduction

When helping a student through problem-solving on a canvas, the tutor may naturally annotate in real time—writing key terms, marking errors or hints, and drawing connections between concepts to complement verbal explanation. This interactive and real-time handwriting-based tutoring supports the understanding of abstract STEM concepts. In particular, the shared canvas keeps intermediate steps visible and offloads working memory [62, 131]. Immediate

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution 4.0 International License. *CHI '26, Barcelona, Spain*

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2278-3/26/04
<https://doi.org/10.1145/3772318.3790935>

feedback with visual annotations guides attention effectively and makes key points salient, which supports faster error correction and smoother conceptual understanding [32, 65, 125].

The inherently interactive nature of handwriting-based tutoring makes it primarily a one-on-one practice and, like other pedagogies that require close tutor involvement, it faces substantial scalability constraints. Traditional in-person education restricts opportunities for individualized learning with high student-tutor ratios and limited infrastructure [28, 59]. The scarcity is visible in university contexts, where long lines form during brief office hours, and even motivated students struggle to obtain timely, one-on-one help. It is simply impossible to provide every student with an individual human tutor for handwriting-based tutoring on demand. Even when tutors are available, handwriting-based teaching demands intensive workloads: tutors must carefully examine each student’s work, interpret their problem-solving approaches, and provide individualized written feedback, a process inherently more time-consuming than standardized methods [81]. These constraints, both the impossibility of universal access and the time-intensive nature of handwriting instruction, inevitably limit who can benefit from this proven pedagogical approach.

Online education has alleviated some general scalability challenges—expanding access across time and place, enabling large cohorts, and automating parts of assessment—but preserving the pedagogical benefits of handwriting remains difficult. Recent online learning platforms have attempted to incorporate some handwriting features to preserve this crucial pedagogical approach. Synchronous platforms such as Zoom and Microsoft Teams have introduced shared digital whiteboards that enable tutors to handwrite in real time [78, 105, 134]. Meanwhile, AI-powered systems such as Photomath and MathGPT now accept handwritten input through cameras or screenshots for problem recognition [38, 99]. However, these approaches face critical limitations: digital whiteboards still require human tutors for real-time handwritten explanations, defeating the purpose of scalability. Automated systems often utilize AI to recognize handwritten input and present typed solutions in an isolated text panel, rather than integrating feedback directly onto the student’s working canvas where the handwriting occurred. No current platform can replicate the real-time, adaptive handwriting interactions of one-on-one tutoring without requiring a human tutor to manually write, annotate, and provide verbal explanations based on student responses [64]. Thus, the challenge of providing scalable, personalized handwriting instruction that adapts to individual student needs remains unresolved in online education [106].

With the rise of Large Language Models (LLMs), researchers and industry are exploring how they can strengthen online learning. LLMs can maintain context over long exchanges and tailor explanations to each learner [58, 120]. Retrieval-augmented generation lets them consult course materials and cite sources, which

reduces hallucinations, and long-context methods allow responses that account for a student’s learning history [68, 71, 88]. Role-playing extends these benefits by letting models act as classmates, peer companions, or tutors to support experiential practice and motivation [49, 72, 122]. Multimodal capabilities let models read sketches, formulas, and speech together, while returning timely feedback [2, 66, 92, 111]. In STEM subjects, recent models show strong problem-solving performance, including on challenging math benchmarks [14, 123]. Together, these advances point to scalable, personalized tutoring with immediate in-canvas feedback.

As a starting point for exploring handwriting-based AI tutoring, we focus on mathematical problem-solving, which is a critical component of STEM education that naturally lends itself to handwriting-based explanations and feedback. The core handwriting tutoring interactions we target in this domain, such as annotations and symbolic writing, are also foundational to many other STEM subjects. Meanwhile, learners across STEM disciplines engage in a wide variety of domain-specific handwriting practices: layering free-body diagrams with vector annotations in physics, constructing multi-step reaction mechanisms with spatially arranged arrows in chemistry, sketching parametric curves and limit behaviors in calculus, or combining circuit schematics with symbolic reasoning in electrical engineering. The breadth and complexity of these heterogeneous practices make it infeasible to exhaustively cover all handwriting-based tutoring scenarios within a single system or study. We leave such domain-specific extensions to future work, with the present study serving as a foundation for exploring how AI tutors can support handwriting-rich learning across STEM.

We present AmlWrite, an LLM-powered AI tutoring system that simulates authentic one-on-one handwriting-based tutoring experiences on tablet devices. Students can naturally write equations and diagrams while solving problems, ask questions when encountering difficulties, and receive immediate personalized guidance. The AI tutor provides verbal explanations while simultaneously writing and annotating directly on students’ work, with handwritten content temporally aligned with the spoken words—mirroring how a human tutor would naturally explain a problem.

To test the usability and effectiveness of our system, we conducted a user study that simulates a math lecture in linear algebra, using two linear algebra topics as a case study. The study follows three stages based on the Gradual Release of Responsibility model [33, 97]: first, the AI tutor demonstrates problem-solving through handwritten examples; then, the tutor and student collaboratively work through similar problems with scaffolded support; finally, the student independently solves problems while the tutor examines and writes feedback on the student’s solution. In the comparative baseline group, participants went through the same three stages with a ChatGPT-style text-based interface.

We propose the following contributions:

- A design space that characterizes handwriting interactions between an AI tutor and a student during one-on-one mathematical problem-solving sessions.
- AmlWrite, an LLM-powered tutor that supports real-time co-speech handwriting for one-on-one math tutoring on tablets, as a step toward scalable handwriting-based tutoring in STEM.

- A within-subjects comparative user study in linear algebra that evaluates the usability, learning experience, and effectiveness of AmlWrite.

2 Related Work

2.1 Handwriting in STEM Education

Handwriting has been fundamental to STEM education for centuries, serving as an essential tool for expressing mathematical reasoning, scientific notation, and engineering designs [12, 76]. In classrooms, tutors work through complex derivations on blackboards while students transcribe and annotate, facilitating active learning through this parallel writing process [83, 100]. Beyond lectures, handwriting enables collaborative problem-solving during office hours and peer discussions, where shared written work provides immediate visual feedback [35]. STEM courses traditionally require handwritten homework submissions, allowing tutors to see students’ problem-solving processes, including crossed-out attempts and marginal notes [61]. Tutors then provide handwritten feedback on these assignments, marking errors and suggesting alternative approaches [80].

Research demonstrates that handwriting plays a crucial cognitive role in learning through its unique neurological and pedagogical benefits. Brain imaging studies reveal that handwriting potentially activates widespread neural connectivity patterns essential for memory formation and learning [117]. In STEM contexts, handwriting is particularly valuable as it forms the foundation of mathematical instruction through “chalk talk”—the universal practice of writing mathematical narratives while explaining concepts—which has emerged as a central pedagogical genre across cultures [4]. Studies of pen-and-tablet use report reduced cognitive load, clearer understanding, and richer tutor-student interaction [65], and pen-based interfaces preferentially support ideation and problem-solving relative to keyboard input [94]. Handwriting proves especially advantageous for graphical problems and visual representations in mathematics courses [109]. The persistence of handwritten work in STEM education thus reflects empirical evidence of its cognitive advantages for developing spatial reasoning, mathematical fluency, and the ability to communicate complex technical ideas through integrated text and visual representations [30, 70].

2.1.1 Handwriting in One-on-One Tutoring. One-on-one tutoring is widely recognized as one of the most effective instructional methods [9, 90, 118]. Beyond personalized pacing and content, much of this effectiveness derives from the dialogic nature of tutoring interactions: students actively construct knowledge by articulating their reasoning, asking questions, and responding to tutor prompts, while tutors diagnose understanding and adapt instruction through contingent, moment-by-moment responses [15, 16, 41]. This dynamic, co-constructed exchange enables learning opportunities that are difficult to replicate in traditional classroom settings. In a one-on-one tutoring session, handwriting frequently occurs within the conversational interactions between tutor and student. Beyond basic step-by-step writing for problem-solving, co-speech annotations are often used to reference content, resolve ambiguity in the dialogue, or direct each other’s attention. Selective writing is also

employed by the tutor to emphasize key concepts or provide hints that guide the student toward the solution [112]. The effectiveness of such co-speech handwriting aligns with multimedia learning principles, which suggest that combining verbal explanations with visual representations enables learners to build richer mental models [75, 95]. The temporal and spatial integration of speech and writing grounds abstract content in concrete visual form, reducing ambiguity and focusing attention [3, 115].

Despite the numerous cognitive and pedagogical benefits of handwriting in STEM education, scaling such approaches to fully realize their potential remains a significant challenge. The primary limitation lies in the resource-intensive nature of handwritten instruction and feedback [124]. Even without considering cost, assigning a dedicated one-on-one tutor to every student is practically impossible. In classroom settings, the traditional model of tutors working through problems on blackboards becomes increasingly challenging to implement as class sizes grow and tutor-to-student ratios expand [64, 102]. Office hours, a more accessible alternative to one-on-one tutoring, are extremely limited regarding tutor availability and allocated time per student. Students often end up waiting in long lines and may ultimately leave without getting their questions answered. Providing handwritten feedback on homework assignments is particularly demanding, as tutors must carefully examine each student’s mathematical derivations, interpret their problem-solving approaches, and write individualized annotations and corrections. This process is inherently more time-consuming than standardized or automated feedback methods [10, 81]. These scalability constraints create inequities in STEM education, as students in under-resourced settings or large classes may have limited access to the rich, interactive handwritten instruction that has proven so beneficial for developing mathematical fluency and problem-solving skills [27, 57, 77].

2.1.2 Handwriting in Online Education. Online education platforms provide students with flexible, accessible, and affordable learning opportunities that transcend geographical and scheduling constraints. Recent platforms have recognized handwriting as essential for STEM learning and made significant efforts to incorporate this modality, yet they struggle to replicate the dynamic handwriting interactions that characterize one-on-one tutoring.

Massive Open Online Courses (MOOCs) frequently rely on pre-recorded video lectures where tutors’ handwritten work is static and non-adaptive, preventing students from receiving personalized visual feedback on their own problem-solving approaches [7]. Learning Management Systems like Canvas and Blackboard support document uploads of handwritten work but lack real-time handwriting interaction capabilities, forcing students to photograph and submit completed work for delayed, asynchronous feedback [13].

Synchronous platforms have made the most progress in incorporating handwriting through digital whiteboards. Zoom, Microsoft Teams, and similar platforms enable tutors to write equations and annotate solutions in real-time during live sessions [78, 134]. However, these systems face fundamental scalability constraints: tutors cannot simultaneously observe and respond to multiple students’ handwritten work, limiting effective interaction to small groups of 5–10 students for handwriting-intensive subjects like mathematics [107]. When scaled to larger classes, these platforms devolve into

one-directional demonstrations where the tutor writes but cannot engage with individual students’ written solutions.

Recent AI-powered platforms have approached handwriting from the input recognition perspective. Photomath, MathGPT, and Microsoft’s Math Assistant can recognize and interpret handwritten technical notation through camera capture or stylus input [38, 79, 99]. Khan Academy partnered with MyScript to enable handwritten equation input in their mobile applications [31]. However, these systems only process handwritten input to generate typed, pre-rendered, or animated responses—they cannot produce adaptive handwritten explanations that adjust to student needs in real-time [25]. This asymmetry creates an unnatural learning interaction where students write but receive only text-based feedback, losing the visual scaffolding that handwritten annotations and step-by-step solutions provide.

The absence of bidirectional, real-time handwriting in online education particularly impacts STEM education, where complex problem-solving requires iterative visual communication. When students work through multi-step derivations or technical problems, they may need tutors who can annotate their work directly, highlight key steps, and demonstrate alternative solution approaches through handwriting [29, 69]. Current platforms’ inability to provide this real-time, adaptive handwritten feedback forces online STEM education to rely on less effective modalities, limiting students’ development of procedural knowledge and problem-solving skills essential for advanced STEM subjects [35, 43]. This gap between the handwriting interactions proven effective in traditional tutoring and the capabilities of current online platforms represents a critical barrier to scaling quality STEM education.

2.2 Large Language Models in Education

Large Language Models (LLMs) have developed several capabilities critical to educational applications. First, they exhibit advanced natural language generation and pedagogical reasoning, which, combined with retrieval-augmented generation (RAG) [68], enable responses grounded in authoritative curricular content while mitigating hallucinations [71]. Second, LLMs demonstrate strong problem-solving abilities in STEM domains—state-of-the-art reasoning models have achieved gold-medal-level performance on International Mathematical Olympiad problems, requiring both creative insight and rigorous logical reasoning [14, 50]. Third, multimodal LLMs such as GPT-4V and Gemini can interpret sketches, handwriting, visual diagrams, and mathematical formulas [37, 92], ranking among the strongest performers on handwriting recognition and math-evaluation benchmarks for error detection, correction, and localization [36, 86, 89, 126]. These capabilities collectively suggest that LLMs can treat students’ handwritten work as a primary object for analysis and feedback.

Building on these capabilities, recent educational systems leverage LLMs to provide personalized tutoring at scale. Kabudi et al. [58] and Nguyen et al. [88] demonstrate that LLMs can dynamically construct personalized learning paths and generate explanations that explicitly respond to a student’s prior performance and interaction history. Meanwhile, prior work shows that LLMs can adopt a variety of human-like roles [49], such as peer [122] or classmate [72] personas, producing conversational behavior that more

closely resembles human interaction and enhancing engagement in simulated classroom settings. For STEM instruction, systems like MathDial [73], ChemTAsk [98], and Abedi et al. [1] leverage LLMs’ reasoning abilities to guide students through complex multi-step procedures while providing detailed solution explanations, thereby enhancing both conceptual understanding and procedural fluency. Sketch- and handwriting-based systems have also emerged: TaleBrush [18] lets authors sketch a protagonist’s “fortune curve” over narrative time and uses that sketch to interactively steer GPT-generated stories. In STEM education, Augmented Math [17] and Augmented Physics [45] transform static textbook pages into explorable explanations by extracting formulas and diagrams from the page and overlaying interactive visualizations directly on top of the original content, which students manipulate through sketch interactions. Interactive Sketchpad [66] uses vision–language models to provide real-time feedback on student drawings for diagram-based mathematics problems. However, in these systems, sketches function mainly as high-level controllers for triggering visualizations or content generation and do not adapt their responses based on students’ prior performance to provide personalized feedback. Moreover, feedback appears in separate textual dialogue and visualization panels rather than as in-situ corrections on the student’s handwriting, which pulls attention away from the act of writing and reduces handwriting to a one-way input channel rather than a shared medium for ongoing interaction. Critically, none of these systems explores temporally synchronized instruction that tightly couples verbal explanations with visual annotations on students’ handwriting—a hallmark of effective human tutoring which, as discussed in §2.1.1, significantly contributes to the pedagogical benefits of one-on-one instruction.

Motivated by these capabilities and the identified gap in temporally synchronized handwriting-based instruction, we propose AmIWrite, an LLM-powered tutoring system that recreates an authentic one-on-one tutoring experience. AmIWrite (1) transforms course materials into structured presentations delivered through an interactive tutor persona, (2) maintains contextual dialogue to generate feedback tailored to students’ questions and handwritten work, and (3) provides immediate in-situ handwritten annotations synchronized with verbal explanations.

3 Interaction Design

3.1 Learning Mathematical Problem-Solving: Linear Algebra as a Case Study

Mathematics underpins a wide range of STEM subjects. In physics, students routinely solve problems involving systems of equations to analyze forces or circuits; in engineering, they manipulate symbolic expressions to model stress, flow, or control systems; in chemistry, they perform quantitative reasoning for reaction rates, equilibria, and stoichiometry. Across these domains, mathematical problem-solving is not only central to conceptual understanding, but also the primary medium through which learners externalize and refine their reasoning. In everyday practice, this work is predominantly done through handwriting: students write multi-step derivations and equations line by line, while tutors mark up students’ written work with corrections, hints, and scaffolded steps directly on the page. These ubiquitous handwriting practices provide a common

substrate for exploring interaction patterns—such as tracking intermediate steps, annotating in context, and spatially organizing work—that can serve as a foundation for future handwriting-based tutoring systems in other STEM areas.

From a technical standpoint, current large language models have shown promising capabilities in mathematical problem-solving, especially for symbolic manipulation and step-by-step reasoning with equations. At the same time, they still face limitations in handling highly complex, unstructured diagrams and rich visual artifacts that are used in some STEM domains (e.g., intricate circuit schematics, multi-layered mechanical assemblies, or densely annotated reaction pathways). Focusing on mathematics allows us to leverage the strengths of LLMs in symbolic reasoning while avoiding, for now, the full complexity of arbitrary visual diagrams. Together with the prevalence of handwritten math work in STEM education, these factors make mathematical problem-solving a natural starting point for exploring scalable handwriting-based tutoring.

Within mathematics, we choose linear algebra as our initial case study domain. Linear algebra is a foundational subject across STEM fields, supporting areas such as computer graphics, signal processing, data science, and, critically, modern AI and machine learning. Moreover, linear algebra problems exhibit rich spatial structure compared to many one-dimensional equation-solving tasks: matrices are inherently two-dimensional objects, where the position of each element carries meaning. This spatial richness allows us to better exploit the image and spatial referencing capabilities of contemporary multimodal LLMs—for example, by pointing to specific matrix entries, highlighting sub-blocks, or annotating particular rows and columns. By designing and evaluating AmIWrite’s handwriting interactions in this linear algebra context, we focus on a mathematically central and spatially expressive domain that both showcases the strengths of current LLMs and provides insights that can extend to other handwriting-intensive STEM topics.

3.2 Co-Speech Handwriting

In human tutoring, tutors naturally combine verbal explanations with handwritten visual elements to enhance comprehension [4]. Drawing from this pedagogical practice, we designed co-speech handwriting interactions where the AI tutor simultaneously speaks and writes, leveraging both auditory and visual channels to support learning. We identified two primary categories of handwriting behaviors: *text writing* and *annotation*.

3.2.1 Text Writing. Text writing encompasses the generation of new textual content alongside verbal explanations. We identified three distinct text writing interactions:

Declarative Writing produces important statements or questions that serve as anchoring points for the lesson. These written elements capture core concepts, learning objectives, or thought-provoking questions (Figure 1a). For example, when introducing matrix rank, the tutor might write “The rank is the number of linearly independent rows” while explaining the concept verbally, providing students with a persistent visual reference for this fundamental definition.

Procedural Writing demonstrates problem-solving procedures (Figure 1h). As the AI tutor explains each step verbally, it writes corresponding mathematical expressions, logical operations, or

procedural instructions. This interaction helps students follow complex reasoning processes by providing a visual trace of the solution pathway that complements the verbal walkthrough.

Selective Writing selectively inscribes critical keywords, operations, or equations mentioned in the verbal content (Figure 1d). Rather than transcribing entire explanations, this interaction highlights essential concepts through strategic writing. For instance, while explaining how to find matrix rank verbally, the tutor might write key terms like “row echelon form,” “pivot,” or “linear independence” to emphasize crucial concepts in the rank-finding process.

3.2.2 Annotation. Annotation interactions modify or emphasize existing content through graphical marks to direct attention and clarify relationships. We identified four annotation types:

Underline highlights single lines or individual elements to draw attention to specific components (Figure 1e). This interaction is particularly useful for emphasizing critical values in equations, key phrases in problem statements, or important results that students should remember.

Circle encompasses regions containing multiple related elements, creating visual groupings that help students recognize conceptual units (Figure 1g). Circles can delineate problem components, group related terms, or isolate areas requiring special attention, helping students parse complex information into manageable chunks.

Arrow establishes explicit connections between elements, illustrating relationships that might not be immediately apparent (Figure 1f). Arrows indicate causal relationships, show transformations between steps, or connect related concepts across different parts of the workspace, making implicit connections visible.

Check/Cross Mark provides visual feedback on correctness (Figure 1c). Check marks validate correct solutions or reasoning steps, while cross marks identify errors. This creates a clear visual language for assessment that students can quickly interpret alongside verbal feedback.

These handwriting interactions can be combined within a single explanation, allowing the AI tutor to create rich multimodal presentations. For example, the tutor might write a mathematical equation (text writing), circle a key term within it (annotation), and then draw an arrow to connect it to a previously written concept (annotation), all while providing verbal explanations. This flexibility enables the tutor to adapt its visual communication to match the pedagogical needs of different learning scenarios.

Temporal alignment and persistence. The temporal alignment between handwriting and speech is critical for effective co-speech interaction. When the tutor mentions “pivot elements” verbally, the corresponding underlining or circling should appear simultaneously or immediately after, creating a cohesive multimodal message. This synchronization reduces cognitive load by allowing students to process related information through both channels concurrently.

3.3 Handwriting Scenarios

Beyond the types of handwriting interactions, we identified three distinct pedagogical scenarios that shape how these interactions are deployed: *lecture*, *guidance*, and *practice*. This progression follows the Gradual Release of Responsibility model [33], transitioning

from tutor modeling (“I do”), to guided instruction (“We do”), to independent practice (“You do”). Each scenario employs different patterns of handwriting interactions to support its specific learning objectives.

3.3.1 Lecture. In the lecture scenario, the AI tutor takes a demonstrative role, presenting concepts and working through sample problems. The tutor employs extensive text writing to build up complete explanations, such as writing out the full definition of matrix rank and then demonstrating the row reduction process on a sample matrix. Declarative writing establishes foundational concepts, while procedural writing creates a comprehensive solution trace that students can review. When students ask questions during lecture, the tutor leverages the existing written content as a visual scaffold, adding annotations to clarify specific points. For instance, if a student asks why certain rows don’t contribute to the rank, the tutor might circle the linearly dependent rows and draw arrows showing their relationship to other rows, while verbally explaining the concept of linear dependence.

3.3.2 Guidance. The guidance scenario positions the tutor as a collaborative problem-solving partner. Here, the tutor presents a problem and guides the student through solving it step-by-step. Initially, the tutor might use minimal writing—perhaps just the problem statement—allowing the student to attempt the first steps independently. When the student encounters difficulties, the tutor progressively increases support through strategic handwriting. Selective writing might highlight important concepts the student should consider, such as writing “pivot” when the student struggles to identify which elements determine the rank. If the student remains stuck, the tutor provides more explicit hints through annotations, such as circling the row that should be reduced next. Should the student continue to struggle, the tutor completes the solution using procedural writing, transforming the interaction temporarily into a mini-lecture that demonstrates the correct approach while maintaining the problem-solving context.

3.3.3 Practice. In the practice scenario, the student works independently before receiving feedback. The tutor initially remains passive, allowing the student to complete their solution without intervention. Once the student finishes, the tutor’s handwriting serves an evaluative function. Check marks validate correct steps, such as proper execution of row operations or accurate identification of pivot positions. Cross marks identify errors, with the tutor then using a combination of annotations and text writing to provide corrections. For instance, if a student miscalculates matrix multiplication, the tutor might cross out the incorrect entry, write the correct value, and use arrows to show which row and column elements should have been multiplied. This scenario emphasizes the use of annotations for assessment, while text writing supplies corrective explanations that help students understand their mistakes.

These scenarios are not rigidly bounded; a single tutoring session can fluidly transition between them based on student needs. For example, a session might begin with a lecture to introduce matrix rank, shift to guidance as the student attempts their first rank calculation, and conclude with practice on additional problems. The handwriting interactions adapt accordingly, with the tutor

modulating the type and extent of visual support to match the pedagogical goals of each scenario.

4 System Overview

4.1 System Walkthrough

The overall system workflow is illustrated in Figure 2 (a–k), demonstrating how students engage with STEM problems through multi-modal interaction on a digital canvas and receive intelligent tutoring feedback via synchronized audio-visual guidance. This workflow is exemplified through a matrix multiplication problem-solving scenario.

The interaction begins when the student writes mathematical equations on the digital canvas using a stylus and subsequently provides voice input stating, "I think I finished writing the answer". The system immediately processes this multi-modal input by converting speech to text (Figure 2a) and capturing screenshots of each page of the canvas (Figure 2b). To provide contextual awareness, the system applies dynamic color-coding to distinguish newly written content in blue from previous attempts (Figure 2c) and overlays a grid coordinate system for precise spatial referencing (Figure 2d).

During initialization, the AmIWrite framework integrates course materials on matrix operations (Figure 2e) and generates pedagogically informed system prompts that incorporate both curriculum content and instructional requirements (Figure 2f).

The core reasoning engine analyzes students' handwritten work and spoken questions based on their historical work and interaction patterns (Figure 2g), generating textual feedback with handwriting function tags embedded with spatial information, such as "Seems we have a [cross; H9] mistake here. The [line; H6, H6] [write; Number of Rows, I2] number of rows of AB is equal to [arrow; H6, D5] the number of rows of [circle; B4, D5] matrix A." (Figure 2h). The system processes these handwriting function tags within the square brackets to generate corresponding visual animations (Figure 2i) while simultaneously converting the associated textual feedback to speech with temporal prediction for each word (Figure 2j). Finally, the system provides synchronized multimodal feedback, coordinating audio explanations with visual annotations triggered by handwriting function tags, and highlighting relevant mathematical expressions on the canvas to achieve seamless integration of auditory and spatial visual guidance (Figure 2k).

In summary, AmIWrite analyzes user handwriting and speech input on the digital canvas, providing history-aware, real-time educational assessment and feedback through temporally-aligned audio explanations synchronized with canvas annotations, enabling seamless integration of auditory instruction with visual learning materials.

4.2 Student Input and Preprocessing

AmIWrite enables student interaction through handwritten input and voice control, replicating authentic classroom dynamics to provide an immersive learning experience.

4.2.1 Voice-Activated System Control. To maintain a smooth user experience and cognitive fluency during learning, the system integrates voice control functionality, enabling users to interact at any

point to simulate natural classroom interactions. The speech recognition system employs an integrated automatic detection mechanism with a Voice Activity Detection (VAD) component (Figure 2a). Specifically, recording begins when the user's voice continuously exceeds the starting threshold for 0.2 seconds and stops when it remains below the ending threshold for 1.5 consecutive seconds. The threshold calculation is based on the probability of human speech detected by the VAD system in each frame, with pre-defined starting and ending thresholds. Subsequently, the system converts audio to text using OpenAI's GPT-4o-Transcribe model [91], and the transcribed text is sent to the reasoning model for further processing.

4.2.2 Handwritten Expression and Intent Recognition. Students can engage with the system through handwritten input at any point during the learning process. When a user completes their handwriting and gives a voice inquiry, the system initiates a three-step canvas processing workflow. First, the system captures screenshots of each page and renames them as "page_{i}.jpg", where i represents the current page number (Figure 2b). Second, the system identifies any new handwriting added by the student since the last system response, converting these additions to blue color to help the reasoning system better locate the student's most recent content (Figure 2c). Finally, the system overlays each screenshot with a purple grid containing 21 rows (labeled A-U) and 13 columns (numbered 1-13) around the periphery (Figure 2d). This grid serves as a positional reference system, enabling the system to provide precise visual feedback and location-specific guidance in subsequent interactions.

4.3 System-Prompt Framework

4.3.1 Automated Lesson Plan Generation. AmIWrite supports multi-domain PDF teaching materials as input (Figure 2e). Upon receiving these materials, the system automatically extracts the curriculum framework using natural language processing and content analysis techniques. Then it generates the three types of structured learning content described in Section 3.3, including quiz exercises and the corresponding solutions. This automated processing results in a comprehensive *Lesson Plan*.

The lesson plan incorporates the six handwriting function tags which are described in Section 4.3.2. During instruction, the system strictly follows this lesson plan and triggers the corresponding visual handwriting animations when key points are spoken in the speech. Furthermore, the system supports [break] tags that pause instruction to confirm student understanding.

The system provides tutors with a preview function to examine the generated learning progression and offers intuitive course refinement tools for curriculum customization. This approach significantly reduces manual curriculum development overhead while maintaining pedagogical quality and ensuring alignment with learning objectives.

4.3.2 Adaptive System-Prompt Engineering. AmIWrite adopts Gemini 2.5 Pro [37] as the reasoning model (Figure 2g).

To control the behavior of the reasoning system, we designed a comprehensive **System Prompt** (Figure 2f) that guides the reasoning logic and teaching behavior, ensuring instruction follows the

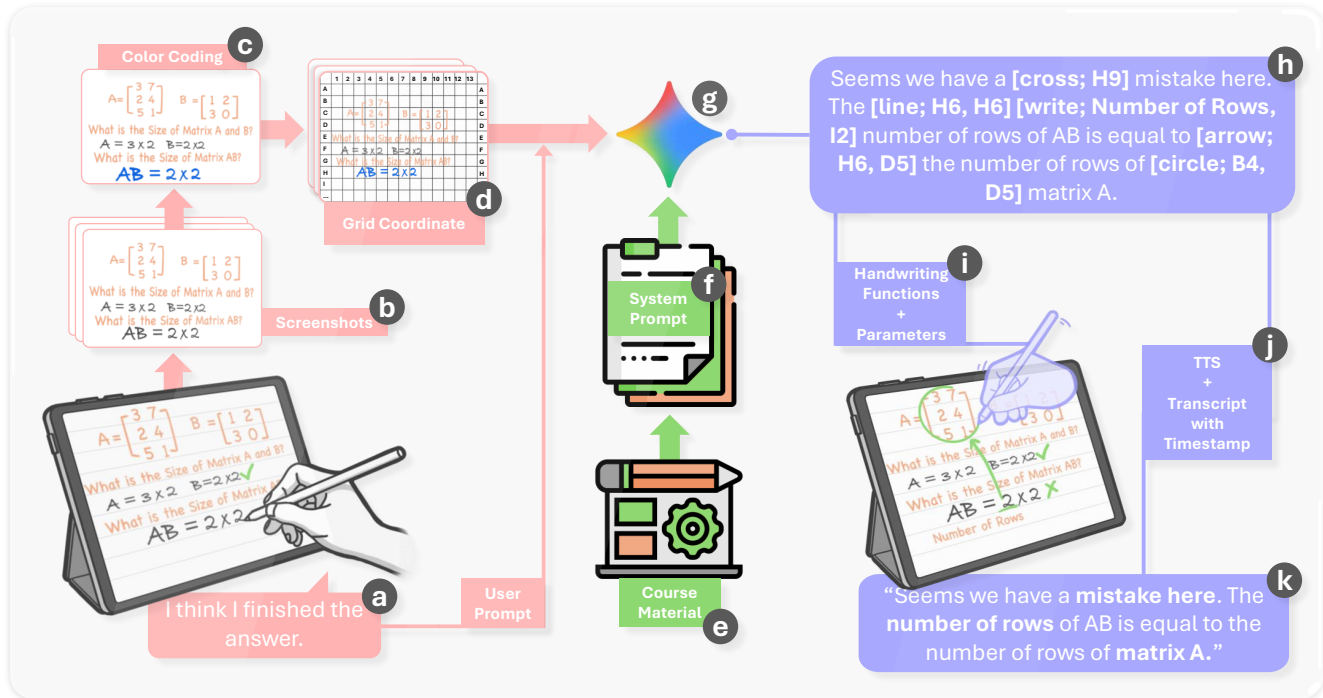


Figure 2: AmlWrite system overview: The system architecture comprises three core processing modules: (Red segment on the left) **Input Processing Module:** (a) speech-to-text transcription, (b) screenshots of all canvas pages, (c) color-coding of the latest student’s handwriting content, and (d) grid coordinate system overlay; (Green segment in the middle) **Knowledge Integration Module:** (e) pre-prepared course material and (f) adaptive system-prompt generation; (Purple segment on the right) **Intelligent Feedback Module:** (g) Gemini 2.5 Pro reasoning engine, (h) textual feedback generation with embedded handwriting function tags, (i) handwriting function tag filter and handwriting animation generation, (j) text-to-speech synthesis with word-level timestamps, and (k) synchronized multi-modal output (writing + speech) to the student’s canvas.

lesson plan described in Section 4.3.1. This framework comprises several key components:

Spatial Reference Prompting: The reasoning system should refer to specific grid coordinates in its responses, following the format [PageRowColumn]. For example, the second page, row C, column 8 is represented as P2C8.

Handwriting Function Prompting: We define six types of handwriting functions and encourage the system to utilize them to clarify speech content when discussing key concepts. The reasoning system returns text strings containing these tags to trigger visual handwriting animations when key locations are reached. For example, when the reasoning system returns "Seems we have a [cross; P1H9] mistake here," AmlWrite draws a cross at Page 1, Row H, Column 9 when the tutorial speech reaches the word "mistake". We defined the following handwriting functions:

- **write:** Displays contextual text at specified locations supporting three pedagogical functions (declarative, procedural, selective). Tag format: [write; text, start_coord]
- **line:** Generates underlines for single-element or single-row highlighting. Tag format: [line; start_coord, end_coord]
- **circle:** Circle around a regional content (ex., a whole matrix). Tag format: [circle; top_left_coord, bottom_right_coord]

- **arrow:** Draw directional arrows to illustrate causal relationships and knowledge transfer. Tag format: [arrow; start_coord, end_coord]
- **check:** Place checkmarks indicating correct processes or results. Tag format: [check; coord]
- **cross:** Place crosses marking incorrect processes or results. Tag format: [cross; coord]

Tutor Behavior Prompting: The AmlWrite system strictly adheres to system prompt guidelines, enabling instructors to modify system prompt and lesson plan based on students’ learning progress and objectives to control the teaching process. In our user study, we prompt the behavior of the AI tutor based on the three instructional scenarios described in Section 3.3:

Lecture: During this phase, the system delivers instruction to students following the lesson plan. The system provides lectures through synchronized visual annotations and speech explanations, and integrates declarative writing with other annotations to present definitions and formulas. Instruction pauses automatically based on [break] tags in the lesson plan to verify student comprehension. Students can interact through voice and handwriting to ask questions—for example, circling formulas or numbers on the canvas to request detailed explanations.

Guidance: In this phase, the system presents a problem and guides students through step-by-step solutions. The system decomposes the problem-solving process into discrete steps according to the lesson plan, focusing on one step at a time. For each step, the tutor suggests the operation needed for the current step using declarative writing, awaits student responses, and provides feedback based on their performance. When the student reports difficulty or makes a mistake, the system offers high-level conceptual hints instead of providing direct answers, encouraging independent problem-solving. The tutor is prompted to provide effective hints by utilizing the handwriting functions. For example, the error mark is used to annotate the location of the mistake. Selective writing is used to emphasize important terms and operations; underlines and circles are used to highlight the elements in the canvas that need attention; and arrows are used to suggest relations between the numbers or key terms. When students repeatedly struggle to solve problems despite all the hints (in AmlWrite, we defined three hint rounds per step), the tutor will then provide a detailed procedural solution and answer. After each step, the tutor confirms with the student whether the student wants to proceed to the next step or still needs further clarification on the current one.

Practice: During this phase, the tutor is prompted to pose a problem from the learning material, which the student must solve independently. Once the student reports completion, the tutor evaluates the student's work, acknowledges the correct solution with a check mark, and annotates each mistake with a cross mark. The tutor then goes through each mistake using various handwriting functions to enhance understanding. The tutor will first explain the mistake, and then write the solution using procedural writing. After each explanation, the tutor checks with the student for follow-up questions.

During the one-on-one session, the student can control instructional progression at any time, including skip, pause, and repeat functions, by simply saying what they need with the speech detection feature.

Please refer to the supplemental materials for the complete system prompt and lesson plan.

4.4 System Feedback Pipeline

4.4.1 Information Extraction from Raw Data. Based on the student's voice and handwriting input, the reasoning model (Figure 2g) analyzes the student's intent following system prompt instructions and returns generated results as a text string with handwriting function tags (Figure 2h). Upon receiving raw text output, AmlWrite filters the string according to the annotation logic mentioned in Section 4.3.2, categorizing all information into visual annotations and speech scripts, as well as the position of each annotation in the speech for further processing. When encountering non-compliant outputs, the system attempts to correct them. For example, if a [write] annotation lacks clearly specified writing positions, the system automatically places it in blank lines immediately following the existing content. If the annotation cannot be corrected, the system skips that annotation.

4.4.2 Annotation Framework and Visual Animations. The system determines when, where, and which animation to render based on the filtered annotation information (Figure 2i). When the animation

references content from other pages, it triggers seamless window transitions. During rendering, each annotation displays a distinct and smooth animation, accompanied by a virtual hand that simulates the writing motion of a real tutor, effectively guiding students' visual attention. The visual effects of each annotation type rendered in the UI are shown in Figure 3.

4.4.3 Synchronized Speech Synthesis. AmlWrite utilizes OpenAI's GPT-4o-Transcribe model [91] to convert filtered speech scripts into individual audio files. To synchronize speech and visual animations, we predict the temporal occurrence of each word in the speech and combine this with annotation localization information obtained from the information extraction process described in Section 4.4.1, enabling timely playback of annotation animations (Figure 2j).

To predict the temporal occurrence of each word in the speech, we implement a lightweight audio alignment workflow that combines duration heuristics with energy-based refinement methods. Specifically, we first estimate word-level temporal positions in the speech proportionally based on punctuation and contextual factors, followed by global scaling to match the total audio duration. The speech timing of each word is then aligned to local minima in the smoothed waveform energy distribution, providing more natural segmentation points. This hybrid approach, which combines text-driven assignment with signal-based correction, provides a computationally efficient solution while achieving sufficient accuracy for real-time annotation and feedback. Finally, the system estimates when the speech reaches relevant annotation content and triggers the corresponding annotation animations, delivering synchronized visual annotations and speech-integrated instructional materials to students (Figure 2k). This approach simulates authentic classroom environments while enabling seamless comprehension of contextual information within the canvas.

4.5 System Setup

The implementation of AmlWrite is a web-based system with an HTML-based front-end user interface and a JavaScript-based back-end server. The back-end server was deployed on a PC (Intel Core i9-10900KF CPU, 3.7 GHz, 128 GB RAM, NVIDIA GeForce RTX 3090), and the interface was tested on an iPad Pro (13-inch, 2nd generation) with Apple Pencil (2nd generation).

5 User Study

We conducted an IRB-approved within-subjects comparative study by exposing participants to two conditions: a) our AI tutor system with full co-speech handwriting capabilities (Figure 4a), and b) a baseline ChatGPT-style conversational tutor that provides text-based explanations without handwriting features (Figure 4b). Participants learned two distinct matrix concepts—matrix rank and matrix multiplication—with each condition teaching one concept. These entry-level topics were selected given their accessibility to participants with minimal linear algebra background. Importantly, they involve non-overlapping procedural knowledge: while rank involves performing row operations to reduce a matrix to echelon form and counting pivot positions, multiplication involves computing dot products between rows and columns. Neither procedure serves as a prerequisite or sub-component of the other, minimizing skill transfer and carryover effects in our within-subjects design.

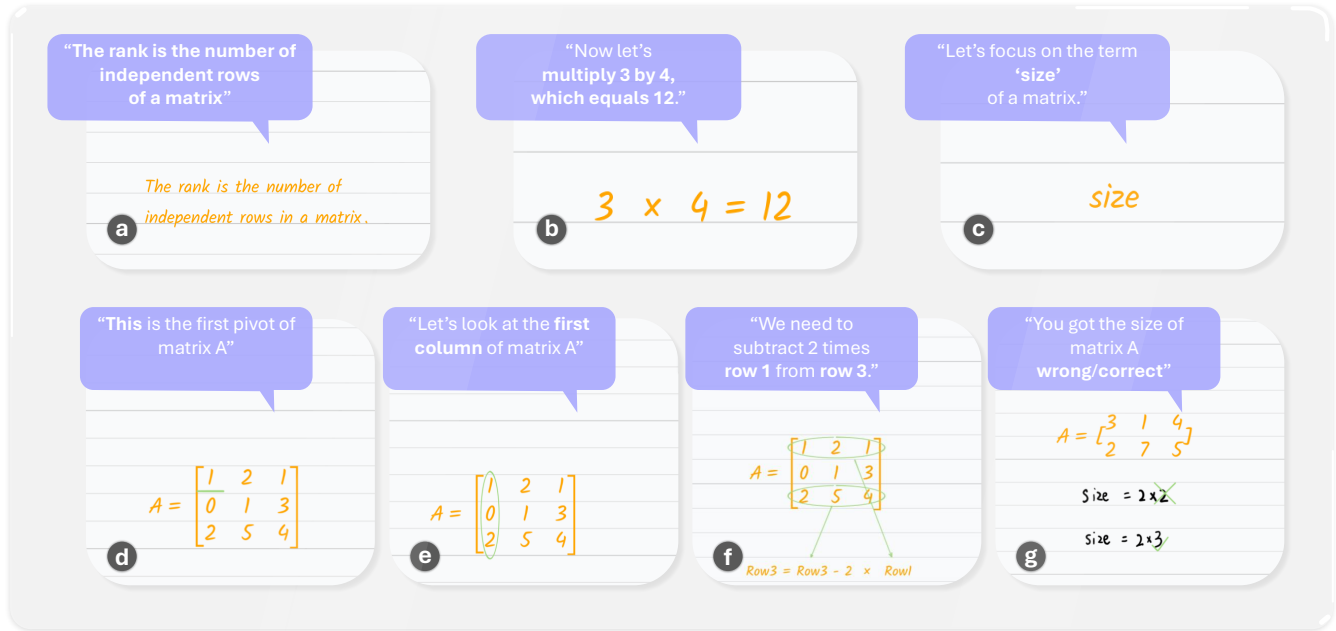


Figure 3: Handwriting features of AmIWrite: a) declarative writing, b) procedural writing, c) selective writing, d) underline, e) circle, f) arrow, and g) check/cross mark.

Both conditions guided participants through the same three pedagogical stages: lecture (concept introduction and demonstration), guidance (collaborative problem-solving with dynamic guidance), and practice (independent problem-solving with assessment and feedback). For the baseline condition, the tutor responded purely through text responses, similar to existing conversational chatbot interfaces. In typical interactions with ChatGPT-style agents, users would manually take a screenshot of their canvas and send it to the agent along with a question. Due to the time constraints of our user study, we automated this screenshot step in the baseline condition so that participants only needed to ask their questions verbally or in text. For our system condition, the tutor employed the full range of handwriting interactions (declarative, procedural, and selective writing with underline, circle, arrow, and correctness mark) synchronized with verbal explanations throughout all three stages. Figure 4 illustrates how the two systems respond when a student makes a subtle sign error in matrix multiplication. The student is computing the (1, 1) entry of $A \times B$ and mistakenly writes $2 \times 1 + 1 \times 0 + 0 \times 5 = 2$, dropping the negative sign on the -1 in the first row of A . In AmIWrite (Figure 4a), the tutor directly annotated on the shared canvas along with the verbal response: it highlights the relevant row of matrix A with a circle (Figure 4a-1), and then rewrites the corrected equation $2 \times 1 + (-1) \times 0 + 0 \times 5 = 2$ beneath the original work (Figure 4a-2). Next, the tutor marked the student's error (Figure 4a-3) and drew an arrow to indicate where the negative sign came from (Figure 4a-4). In the baseline ChatGPT-style condition (Figure 4b), the agent returns a paragraph of text explaining that the first row is $[2, -1, 0]$ and that the sign should be negative. Unlike AmIWrite, this feedback is not spatially

aligned with the canvas, requiring the student to mentally map the textual description back onto their handwritten steps.

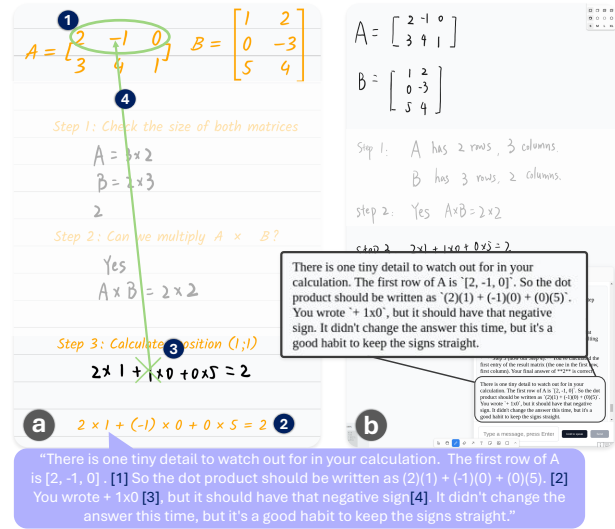


Figure 4: System interfaces for user study. a) AmIWrite, b) Baseline.

5.1 Procedure

5.1.1 Setup. Before the study, participants reviewed and signed an informed consent form and completed a demographics questionnaire. The main sessions were conducted in a quiet room; the

participants used a 13-inch iPad Pro and Apple Pencil for the learning experience. Before each condition, participants received a 2-minute tutorial on the interface and were told they would progress through a lecture, guidance, and practice sequence. We used a within-subjects, fully counterbalanced crossover design. Each participant completed two modules (matrix rank and matrix multiplication), one with our system and one with the baseline. The four sequences were: (A) rank with our system then multiplication with the baseline; (B) rank with the baseline then multiplication with our system; (C) multiplication with our system then rank with the baseline; and (D) multiplication with the baseline then rank with our system. To reduce carryover effects, a 5-minute break with a simple word-puzzle distractor separated the two learning modules [44, 103].

5.1.2 Learning Content and Structure. Each learning module followed an identical three-stage structure lasting approximately 25 minutes:

Lecture Stage (~10 minutes): The tutor introduced the core concept through explanation and demonstration. For matrix rank, the lecture covered the definition of rank, linear independence, and the process of reducing rows to the echelon form. For matrix multiplication, the lecture included dimensional requirements, the size of the product, and the row-by-column multiplication rule. Both conditions delivered identical lecture content on the canvas. The key difference emerged when participants asked questions: in the handwriting condition, the tutor responded verbally while adding co-speech annotations and writing text to the canvas, whereas in the baseline condition, responses appeared as text messages in the textbox alongside the canvas content.

Guidance Stage (~10 minutes): Participants attempted a sample problem with graduated support from the tutor. When participants struggled, the tutor provided progressively specific hints and corrections, starting with conceptual reminders (“What makes a row linearly independent?”) and advancing to procedural guidance (“Try eliminating the first element of row 2”). In the handwriting condition, hints combined verbal explanations with handwritten support, where keywords were written on the canvas using selective writing, and annotations (circles, arrows, underlines) were used to highlight relevant matrix elements or relationships. The baseline condition delivered equivalent hints as text messages in the chatbox rather than visual markings. After three unsuccessful attempts, the tutor demonstrated or completed the procedure.

Practice Stage (~7 minutes): Participants independently solved another problem at this stage. They could not request hints during problem-solving, but received detailed evaluation afterward. In the handwriting condition, the tutor marked correct steps with check marks, incorrect steps with crosses, and provided visual corrections showing the proper approach along with explanations. The baseline condition provided textual feedback listing correct and incorrect steps with explanations.

The lecture scripts and problem sets are available in the supplemental materials.

5.1.3 Evaluation Metrics. We collected both quantitative and qualitative measures to assess learning effectiveness and user experience across four categories: process measures, learning outcomes, subjective measures, and qualitative data.

Learning Outcomes: Each module included pre- and post-tests consisting of 10 questions. From these tests, we calculated learning gain to account for differences in prior knowledge. To control for order effects, the order of equivalent test forms was counterbalanced across participants and modules (e.g., Form A as pre-test and Form B as post-test, or vice versa). The pre- and post-test templates are available in the supplemental materials.

Subjective Measures: After completing each module, participants answered a 7-point Likert-scale questionnaire assessing their subjective experience with the system’s key interaction features. Following established practice in HCI systems research [48, 84], items were designed to capture user perceptions of system-specific aspects not addressed by standardized usability scales, including comprehension of AI tutor instructions, ability to locate referenced content, awareness of errors, attention demands when switching between instruction and practice, and perceived naturalness of communication.

We also administered the NASA Task Load Index (NASA-TLX) to assess cognitive load across six dimensions, providing insights into the mental demand of learning with each system [46]. Additionally, participants completed a System Usability Scale (SUS) survey for each condition to measure perceived usability and satisfaction [11].

Process Measures: We tracked help-seeking behavior by recording the number and types of inquiries requested during the study, which provided insights into when and how students struggled with the learning material and sought to interact with the AI tutor.

Qualitative Data: We conducted 10-minute semi-structured interviews after participants completed both conditions, exploring their preferences, perceived advantages and disadvantages of each system, and suggestions for improvement. Throughout the study, we captured screen recordings of all interactions, enabling post-hoc analysis of system usage patterns. These recordings revealed how participants navigated each interface and adapted their learning strategies to the different modalities.

5.2 Participants

We recruited 40 participants from the university (20 male, 20 female), ranging from undergraduate to graduate students ($M = 25.0$, $SD = 3.13$). More than half of the participants (55%) reported having used online learning platforms such as Coursera, Khan Academy, Canvas, YouTube, EdX, and Duolingo. A large majority (88%) had previously used large language models (LLMs) such as ChatGPT or Gemini in their courses.

In terms of academic background, 78% of the participants majored in engineering fields (e.g., Mechanical Engineering, Electrical Engineering, Civil Engineering, Industrial Engineering, Materials Science, Aeronautical Engineering Technology, Computer Science), while 22% came from non-engineering disciplines (e.g., Psychology, Public Health, Chemistry, Management, Film, Environmental Studies). Regarding mathematics preparation, 57% of the participants reported intermediate undergraduate coursework, 25% advanced undergraduate or graduate-level coursework, 12% introductory or high school-level coursework, and 5% were not familiar. With respect to linear algebra, 70% reported having entry-level knowledge, 5% reported intermediate-level knowledge of matrix operations and

solving systems, and 25% reported being unfamiliar. No advanced-level or expert-level users were involved in the study (0%).

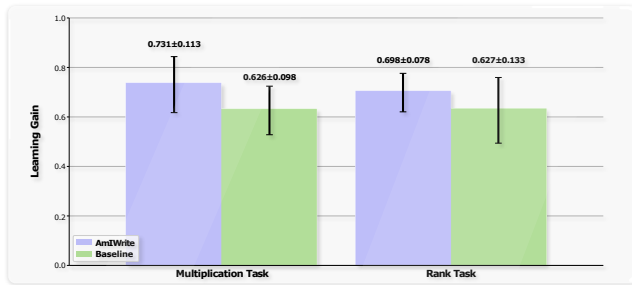


Figure 5: Pre- and post-test learning gain

5.3 Result Evaluation

5.3.1 Quantitative Results. Each participant completed (i) a feature-focused experience questionnaire, (ii) the NASA Task Load Index (NASA-TLX), and (iii) the System Usability Scale (SUS). In addition, each learning module included pre- and post-tests from which we computed learning gain. Shapiro-Wilk tests indicated non-normal distributions; therefore, Wilcoxon signed-rank tests were used for within-subject comparisons across conditions (questionnaires), and Mann-Whitney U test was used for between-subject comparisons (learning gains).

Learning outcomes (Figure 5). Across both modules, participants achieved moderate-to-high learning gains in both conditions; however, nonparametric tests detected no significant differences between our system and the baseline. For the matrix multiplication task, AmIWrite showed a higher mean gain while the difference was not significant ($M_{\text{AmIWrite}} = 0.731$, $SD = 0.506$; $M_{\text{Baseline}} = 0.626$, $SD = 0.439$; $p = 0.464$). For the matrix rank task, learning gains were nearly identical ($M_{\text{AmIWrite}} = 0.698$, $SD = 0.348$; $M_{\text{Baseline}} = 0.627$, $SD = 0.593$; $p = 0.934$).

System feature questionnaire (Figure 6). While learning gains were comparable across conditions, AmIWrite showed significant improvements in user experience across multiple dimensions relative to the baseline. Participants reported easier extraction of key points when using AmIWrite (Q1: $M_{\text{AmIWrite}} = 6.30$, $SD = 0.69$ vs. $M_{\text{Baseline}} = 4.50$, $SD = 1.81$, $p < 0.001$). "When the tutor wrote down something during the lecture, I knew it was important." (P03) The content referenced by the AmIWrite tutor can be more easily located (Q2: $M_{\text{AmIWrite}} = 6.12$, $SD = 1.24$ vs. $M_{\text{Baseline}} = 4.35$, $SD = 1.97$, $p < 0.001$). "The arrow and the circle really help me figure out the corresponding elements." (P07) Participants can better locate their errors from the handwriting feedback (Q3: $M_{\text{AmIWrite}} = 6.40$, $SD = 0.71$ vs. $M_{\text{Baseline}} = 4.92$, $SD = 1.73$, $p < 0.001$). "Compared to find my mistake from the text response, the cross mark is really straightforward." (P07). They found AmIWrite more able to provide the proper information (Q5: $M_{\text{AmIWrite}} = 6.42$, $SD = 0.75$ vs. $M_{\text{Baseline}} = 5.53$, $SD = 1.55$, $p < 0.001$) while avoiding redundant responses (Q6: $M_{\text{AmIWrite}} = 3.45$, $SD = 1.96$ vs. $M_{\text{Baseline}} = 4.50$, $SD = 1.83$, $p < 0.05$). "I feel like with those handwriting features, the responses from the AI are more concise." (P11) The instructions

from AmIWrite were easier to understand (Q7: $M_{\text{AmIWrite}} = 6.45$, $SD = 0.71$ vs. $M_{\text{Baseline}} = 4.97$, $SD = 1.76$, $p < 0.001$) and remember (Q8: $M_{\text{AmIWrite}} = 6.05$, $SD = 1.04$ vs. $M_{\text{Baseline}} = 5.17$, $SD = 1.69$, $p < 0.01$). "I don't need to browse the bulky text response and extract the necessary information; I just look, listen, and remember the instruction." (P21). Communication with the AmIWrite tutor felt more natural (Q9: $M_{\text{AmIWrite}} = 5.33$, $SD = 1.69$ vs. $M_{\text{Baseline}} = 4.45$, $SD = 1.84$, $p < 0.01$). "It's quite similar to an office hour, where the TA guides me on how to solve a problem." (P08) And interactions within AmIWrite were more engaging (Q10: $M_{\text{AmIWrite}} = 5.85$, $SD = 1.29$ vs. $M_{\text{Baseline}} = 4.62$, $SD = 1.98$, $p < 0.001$). "I never had such an interesting online learning experience before." (P03) We observed significant difference in attentional switching demands between the tutor's explanations and participants' ongoing work between the two conditions (Q4: $M_{\text{AmIWrite}} = 2.83$, $SD = 1.50$ vs. $M_{\text{Baseline}} = 3.90$, $SD = 2.05$, $p < 0.01$). "With AmIWrite I could mostly stay on the canvas and follow the handwriting; with the baseline I kept switching my attention between the long text and my work." (P27)

NASA Task Load Index (Figure 7). Temporal demand ($M_{\text{AmIWrite}} = 20.25$, $SD = 22.64$ vs. $M_{\text{Baseline}} = 27.62$, $SD = 27.90$, $p > 0.05$) did not differ significantly between conditions, which is expected because the learning pace was largely self-controlled in both conditions. The rest of the TLX results echo the feature-survey findings. Mental demand ($M_{\text{AmIWrite}} = 29.50$, $SD = 25.69$ vs. $M_{\text{Baseline}} = 43.12$, $SD = 30.23$, $p < 0.01$) was significantly lower for AmIWrite compared to the baseline. AmIWrite substantially reduced physical demand since the users interact with the system without any extra UI elements ($M_{\text{AmIWrite}} = 14.00$, $SD = 14.68$ vs. $M_{\text{Baseline}} = 25.25$, $SD = 28.91$, $p < 0.01$). The participants also reported better perceived performance ($M_{\text{AmIWrite}} = 21.25$, $SD = 24.90$ vs. $M_{\text{Baseline}} = 34.75$, $SD = 30.95$, $p < 0.01$) that required less effort ($M_{\text{AmIWrite}} = 24.50$, $SD = 20.15$ vs. $M_{\text{Baseline}} = 40.00$, $SD = 29.55$, $p < 0.05$), and less frustration ($M_{\text{AmIWrite}} = 16.12$, $SD = 19.89$ vs. $M_{\text{Baseline}} = 27.50$, $SD = 28.49$, $p < 0.01$).

System Usability Scale. SUS evaluation confirmed higher usability for AmIWrite. AmIWrite achieved significantly higher usability scores compared with the baseline ($M_{\text{AmIWrite}} = 78.38$, $SD = 16.03$ vs. $M_{\text{Baseline}} = 63.44$, $SD = 24.22$, $p < 0.001$). The results also indicate AmIWrite achieved "Good" usability.

Carryover Effect Evaluation. To assess potential carryover effects, we first examined whether the order of learning topics (Matrix-first vs. Rank-first) influenced the results. Mann-Whitney U tests comparing the two order conditions revealed no significant effects on learning gains ($p > 0.32$), questionnaire responses ($p > 0.08$), or NASA-TLX ratings ($p > 0.21$). These results confirm that the counterbalanced design successfully controlled for topic order effects.

We further compared the difference scores (AmIWrite – Baseline) between participants who used AmIWrite first (A-first, $n = 20$) versus those who used Baseline first (B-first, $n = 20$) using Mann-Whitney U tests. 14 out of the 17 quantitative measures examined (82%) showed no significant order effects ($p > 0.05$), supporting the validity of our counterbalanced design.

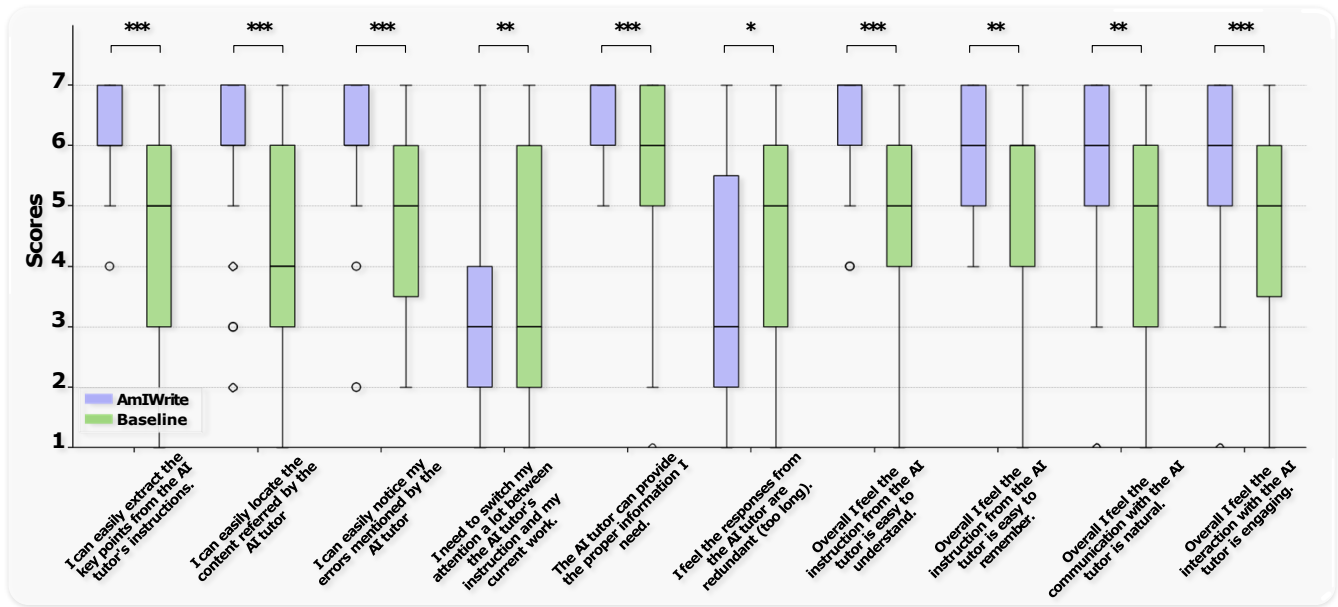


Figure 6: System feature questionnaire results. Significance: * $p < .05$, ** $p < .01$, *** $p < .001$.

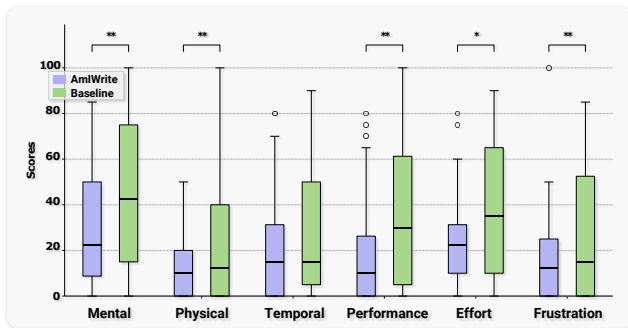


Figure 7: NASA TLX results. Significance: * $p < .05$, ** $p < .01$.

For Q7 (instruction understandability), A-first participants showed a larger preference for AmIWrite ($\Delta = 2.15$) compared to B-first participants ($\Delta = 0.80$, $p = 0.020$). Similarly, for Physical Demand, A-first participants reported a greater reduction with AmIWrite ($\Delta = -19.00$) than B-first participants ($\Delta = -3.50$, $p = 0.048$). These patterns suggest a primacy effect, where initial exposure to AmIWrite's in-situ guidance established a reference point that amplified perceived differences in instruction clarity and physical demand when subsequently experiencing the baseline.

Interestingly, Q6 (response redundancy) showed the opposite pattern ($p = 0.002$): B-first participants rated AmIWrite as more redundant ($\Delta = -2.35$) than A-first participants ($\Delta = 0.25$). This contrast effect aligns with participant feedback that reading text is faster than listening to verbal responses: participants who first adapted to Baseline's text-based delivery may have perceived AmIWrite's audio guidance as comparatively slower-paced and thus

more redundant: "I feel going through the text messages is quicker." (P11)

5.3.2 Observational Results. We logged all interactions between participants and both systems. In the baseline condition, answers appeared in a separate chat panel; in our system, the tutor explained verbally while writing and annotating directly on the canvas.

Response Accuracy. We evaluated AMIWRITE on 690 annotated student steps across two dimensions: (i) *response accuracy*—whether the tutor correctly recognized context and generated appropriate feedback—and (ii) *localization accuracy*—whether the visual annotation was placed in the correct location on the canvas grid. The system achieved 98.8% response accuracy (682/690 correct), with 8 incorrect-feedback cases, all false negatives (i.e., correct work flagged as incorrect due to misinterpretation). Typical recognition failures included symbol misreads due to character similarity or unclear handwriting from the student (7 out of 8 failures). For example, three failures occurred when users positioned the negative sign too close to the number 1, causing "−1" to resemble "4". Additional cases involved users writing near the canvas boundary, resulting in compressed and distorted numbers. In another case, the user directly overlaid a "0" on top of a "3" to make a correction, hindering the system from recognizing the correct number. Localization accuracy was 92.9% (641/690), with 49 mislocalizations (i.e., the annotation is not written at the desired location). To better evaluate the causes of hallucination and recognition error, we conducted a more comprehensive performance evaluation in the next section.

Participant interaction. In the *lecture* stage, behavior was similar across conditions: participants asked clarifying questions about concepts they found difficult. In the *practice* stage, after the tutor delivered full feedback on the participants' solutions, almost all participants ended the session without follow-up questions;

only one participant in the baseline condition asked an additional question.

The clearest contrast emerged in the *guided practice* stage, where the agent's ability to access and annotate the shared canvas became critical. In the AmiWrite condition, the tutor scaffolded each sub-step by first writing relevant content directly on the canvas—for example, extracting and positioning the specific row and column vectors involved in a matrix multiplication step—allowing participants to immediately engage with the computation. In contrast, participants in the baseline condition had to manually transfer every element from the text-based instructions to their canvas, adding substantial overhead before they could begin problem-solving. Furthermore, because the baseline tutor could only reference canvas locations through text, participants frequently scrolled up and down within the chatbox, continuously switching attention between the text dialogue and their canvas to locate the referenced content. This attentional fragmentation was largely absent in the AmiWrite condition, where spatial references were grounded directly through in-situ annotations.

5.3.3 Interview Results.

Natural and Engaging Learning Experience. Participants consistently described the handwriting-based interactions as creating a more authentic and immersive learning environment. The combination of verbal explanations with synchronized handwriting—a core contribution of our temporally-aligned multimodal instruction design—created a sense of presence that participants found both *"natural"* and *"engaging"*. One participant noted: *"It's like a real one-on-one learning session—I listen to the tutor, I raise my questions, the tutor responds and writes on the canvas. I write my answer, and the tutor gives me feedback and annotates on my work immediately."* (P01) This reflection directly speaks to our bidirectional handwriting interaction model, where both tutor and student contribute to a shared canvas, mirroring the dynamic exchange of real-life one-on-one tutoring. The design fostered sustained engagement, with another participant observing: *"It makes me more focused as I am taking a real lesson with a real tutor. If I study by myself, I can easily get distracted."* (P08) Participants also appreciated the in-situ annotation approach compared to text-based dialogue: *"It feels much easier consuming the instructions—I don't need to switch back and forth trying to figure out which part of the canvas the text response is referring to."* (P33) This feedback validates our design choice to spatially ground tutor feedback directly on student work, reducing the cognitive load of cross-referencing between verbal descriptions and canvas locations. Finally, participants raised an interesting point regarding reduced social pressure afforded by the AI tutor modality: *"I feel more comfortable asking questions to the AI tutor—in real life I sometimes feel hesitant or intimidated asking my TA."* (P29) This suggests that AI-mediated tutoring may lower affective barriers to help-seeking by providing a low-stakes environment for students to practice and make mistakes.

Desires for Adaptive Personalization. While appreciating the system's capabilities, participants expressed varied preferences for instructional style and pacing. One participant suggested a less authoritative approach: *"I feel a little bit of pressure from the tutor;*

maybe it can act as a peer and we learn together." (P09) Others highlighted the need for adaptive explanation depth, with contrasting preferences: *"Sometimes I want the tutor to explain the procedure with more detail without me explicitly asking for it."* (P12) while another noted, *"Sometimes I feel the tutor explains the procedure with too many details."* (P24) These divergent preferences underscore the importance of personalized pedagogical adaptation.

Technical Improvements and Accessibility. Participants identified specific technical enhancements that would improve usability. Several participants requested supplementary text captions: *"Sometimes I miss a part from the verbal response so I have to ask the tutor again. It would be helpful if the response could also be displayed as text on the side so I can look at it anytime."* (P11) Response latency emerged as another concern: *"It could feel more natural and realistic if the AI tutor could respond right away."* (P20) Meanwhile, some participants express a neutral attitude towards such latency: *"Even real TA needs to think for a while before giving me suggestions or corrections."* (P25) Regarding recognition errors, since all errors were false negatives during the study, participants did not report significant decreases in trusting the AI tutor: *"At one step it said I was wrong, but while the AI was explaining, I noticed I was actually correct. It wasted a bit of my time, but it didn't affect my learning outcome."* (P17). However, participants emphasized that avoiding false positives was critical: *"If I'm wrong at a point but the system fails to point it out, it will seriously affect my trust."* (P17) The system's localization errors also caused certain confusion: *"Although I could get the idea from the tutor, it was a bit confusing when it circled the blank area next to the reference number."* (P07) Participants highlighted that localization errors can be significant in high-precision tasks: *"If I'm working on a free-body diagram and the tutor highlights the wrong vector, that would throw me off completely."* (P29) Participant also mentioned the limitation of using AmiWrite in quite a public environment: *"Although it feels engaging by talking, I cannot use the system in a library, where I used to prepare for exams."* (P33)

Envisioned Applications. Participants enthusiastically described potential use cases spanning formal and informal learning contexts. One participant envisioned replacing traditional lectures entirely: *"If the system is more mature, I would imagine I no longer need to go to any lectures, because I can have such one-on-one sessions anytime!"* (P02) Others identified specific applications including exam preparation: *"I definitely want to use this system to practice my weaknesses before physics exams"* (P05) and project development: *"Maybe the tutor can support me in developing the system structure chart on the fly for my current prototype app in my CS class"* (P04).

6 Performance Evaluation of Handwriting Recognition

To evaluate the robustness of our handwriting recognition pipeline beyond the user study context, we conducted a performance evaluation using a broader range of linear algebra problems featuring diverse mathematical notations. This evaluation assesses the system's recognition accuracy across varying handwriting styles, notations, and problem complexity.

6.1 Data Collection

We invited a former teaching assistant with linear algebra instruction experience to design 10 fundamental problems with procedural solutions. These problems covered commonly used notations and operations, including various matrix representations, subscripts and superscripts, and systems of linear equations. We recruited 16 participants, each completing all 10 problems by handwriting on the canvas, yielding a total of 160 handwritten responses. To evaluate recognition performance under both correct and incorrect response scenarios, participants were evenly divided into two groups with counterbalanced assignments: Group A wrote correct answers for Questions 1–5 and intentionally incorrect answers for Questions 6–10, while Group B followed the reverse assignment. For correct responses, participants copied the provided procedural solution; for incorrect responses, they introduced at least one error (e.g., numerical mistakes, symbol transcription errors, or formula misapplication) while maintaining structural similarity to the reference answer. This design enables us to assess not only the system’s transcription accuracy but also the LLM’s capability to analyze and identify errors in handwritten mathematical content within a structured problem-solving context.

6.2 Gemini Inference Procedure

For each of the 160 handwritten responses, we simulated our system’s inference setting by issuing a prompt with the same logical structure as in the system and providing Gemini 2.5 Pro with (1) the corresponding reference solution as learning material, and (2) the student’s handwritten answer in our canvas. The model was asked to output a binary correctness judgment, a brief natural-language justification, and the location of the mistake (in the same format as in our user study system), allowing us to assess its ability to perform context-aware handwriting recognition. To mitigate stochasticity, we executed the Gemini 2.5 Pro inference script five times for each response and aggregated the resulting correctness labels, yielding 800 model predictions in total (160 handwritten responses \times 5 runs).

6.3 Results

Table 1: Confusion matrix and derived metrics for Gemini 2.5 Pro’s binary correctness classification on 800 model responses.

Confusion Matrix		Classification Metrics	
TP (True Positive)	372	Accuracy	0.926
TN (True Negative)	369	Precision	0.923
FP (False Positive)	31	Recall	0.930
FN (False Negative)	28	F1-Score	0.927

Table 1 reports the confusion matrix and standard classification metrics for Gemini 2.5 Pro on our handwriting dataset. These values are computed over 800 model predictions in total, treating each run as an independent binary correctness judgment. Table 2 characterizes response-level stability, reporting the distribution of handwritten responses by the number of correct model responses

Table 2: Distribution of the number of correct model responses across five runs for each of the 160 handwritten responses, along with inter-run agreement measured by Fleiss’ κ coefficient.

Correct frequency across 5 runs per handwritten response	
# correct model responses	# user responses
5	135
4	10
3	6
2	3
1	2
0	4
Fleiss’ κ (5 runs/response)	
0.872	

across five runs (0–5), and summarizing inter-run agreement with Fleiss’ κ .

Across these 800 predictions, the LLM correctly classified the vast majority of handwritten answers (accuracy = 0.926 and F1 score = 0.927), indicating that it has sufficient context-aware handwriting recognition and symbolic interpretation capabilities for our basic linear algebra setting. Errors are relatively balanced between false positives and false negatives, and the frequency distribution in Table 2 shows that 135 of the 160 (84.4%) responses received no incorrect prediction across all five runs, with only a small subset (5.6%) accumulating more than two errors. The resulting Fleiss’ κ of 0.872 indicates high agreement across the five repeated runs per response, suggesting that Gemini’s correctness judgments are stable rather than stochastic. The failures are typically associated with small, crowded, or messy handwriting, leading to local rather than global misjudgments of answer correctness.

To further assess localization quality, we examined the model’s performance across all True Negative samples and found 21 cases (5.7%) where Gemini’s predicted location did not align with the actual mistake in the student’s work. This failure proportion is comparable to that observed in the user study (7.1%). For these 21 cases, we labeled the actual location of each mistake and treated these locations as ground truth, then computed the Euclidean distance between Gemini’s predicted locations and the corresponding ground truth locations. The average distance was 1.29 grid cells (SD = 0.55), and in most cases the predicted error location was displaced by only a single cell (16/21, 76.2%). We did not observe any systematic relationship between these localization errors and specific users or handwriting legibility, suggesting that they are better interpreted as occasional hallucination of the underlying LLM.

6.4 Analysis of Error Types

Through our evaluation, we identified two primary categories of system errors: *recognition errors* arising from handwriting interpretation and *hallucinations* stemming from LLM reasoning failures. Figure 8 illustrates representative examples of each category.

6.4.1 Recognition Errors. Recognition errors occurred when the vision-language model failed to accurately interpret handwritten content on the canvas. We identified four common causes:

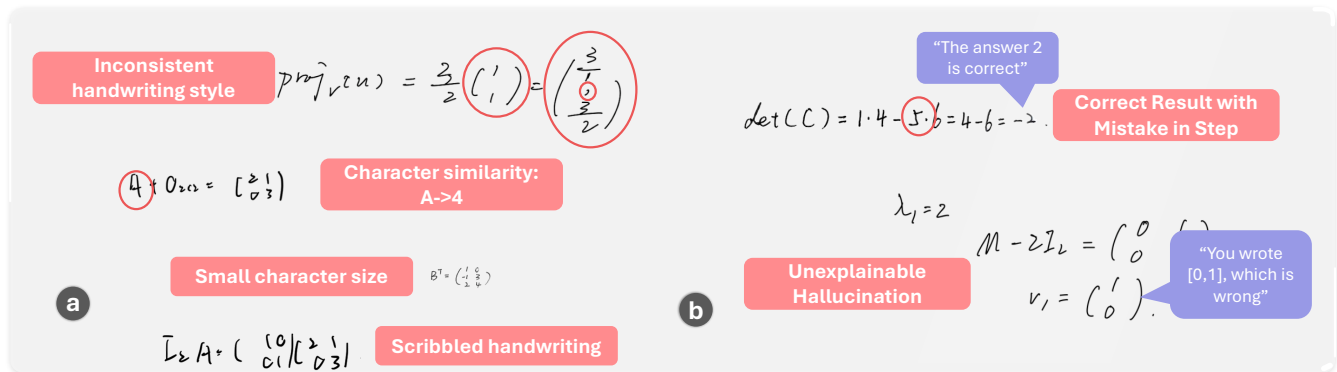


Figure 8: Errors found in current LLM-based handwriting recognition: a) Errors due to handwriting quality: inconsistent handwriting style, character similarity, small character size, and scribbled handwriting; b) Hallucination errors: mistake in the step with the correct answer and unexplainable hallucination.

Inconsistent handwriting style. When participants varied their notation style within a single solution, the model occasionally misinterpreted the content. As shown in Figure 8(a), one participant wrote vectors using inconsistent formatting—one vector without commas between elements and another with commas (e.g., $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ vs. $\begin{pmatrix} 3/2, \\ 1/2 \end{pmatrix}$). Such notational inconsistencies, while easily interpretable by human readers, introduced ambiguity for the LLM and led to recognition errors.

Character similarity (8/59, 13.6%). Visually similar characters posed significant challenges. The model confused characters such as A and 4, 1 and l, 0 and O, or 5 and S. In one instance (Figure 8(a)), the letter A was misrecognized as the digit 4, altering the mathematical meaning entirely.

Small character size (13/59, 22.0%). When participants wrote in reduced scale—common when fitting matrices or multi-step derivations into limited canvas space—recognition accuracy degraded.

Scribbled handwriting (11/59, 18.6%). Rushed or carelessly written strokes may result in recognition failures. Overlapping strokes and incomplete character formation compounded interpretation difficulty.

6.4.2 Hallucinations. Beyond recognition errors, we observed hallucinations where the LLM generated incorrect assessments despite accurate perception of the canvas content:

Correct result with erroneous intermediate reasoning (12/59, 20.3%). In some cases, the tutor correctly identified a final answer but fabricated or miscalculated intermediate steps. As shown in Figure 8(b), the tutor affirmed that the determinant result of -2 was correct, yet the annotated intermediate calculation displayed an error (3×3 instead of 3×2). This inconsistency could reinforce procedural misconceptions even when the student’s final answer is valid.

Unexplainable hallucination (15/59, 25.4%). Occasionally, the tutor incorrectly flagged correct student work as erroneous without valid justification. Figure 8(b) shows an instance where a student

correctly computed an eigenvector as $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ with neat writing, yet the LLM asserted “You wrote $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, which is wrong”—a hallucination with no apparent cause from either recognition failure or reasoning error. Such false negatives risk undermining student confidence and trust in the system.

7 Discussion, Limitations and Future Work

7.1 Enrich Interactive Learning Process through LLMs

Since the public release of ChatGPT in late 2022, LLMs have rapidly permeated educational environments. People now routinely use these tools to obtain quick explanations and worked-out solutions with minimal effort. While this widespread adoption highlights how convenient and efficient LLM tools are for producing ready-made answers, it also encourages a result-oriented mindset in which users may prioritize getting answers over engaging in the intermediate reasoning processes [104]. Consequently, such a result-seeking mentality has introduced concerning patterns of misuse in learning. The ease of obtaining complete solutions from ChatGPT and similar tools has led to widespread academic integrity issues, with students submitting AI-generated work without engaging in the learning process [8, 20, 47]. More troublingly, over-reliance on LLMs for problem-solving can inhibit the development of critical thinking skills and deep conceptual understanding. When students bypass the struggle of working through problems, the very process that builds mathematical reasoning and problem-solving abilities, they miss essential learning opportunities [128, 130].

The challenge, therefore, is not whether to use LLMs in education, but how to leverage LLMs to enrich the interactive learning process rather than simply providing results. Recent research has demonstrated innovative pedagogical approaches to utilize LLMs. Liu et al. [72] developed ClassMeta, where LLM-powered agents embody active student personas in VR classrooms, leveraging peer influence to promote participation and showing how LLMs can simulate beneficial social dynamics in learning environments. Jin

et al. [56] took a different approach by using LLMs as teachable agents that students must instruct, reversing the traditional tutoring dynamic to help learners identify their own knowledge gaps through the act of teaching. Wang et al. [121] created GenMentor, a multi-agent framework that maps learning goals to skills and generates personalized learning paths, demonstrating how LLMs can orchestrate complex, goal-oriented educational experiences. AmIWrite contributes to this emerging paradigm by leveraging LLMs' generative capabilities to create multimodal tutoring interactions that were previously impossible without human tutors. By generating synchronized handwritten explanations alongside verbal guidance, our system maintains the cognitive benefits of traditional handwriting-based tutoring while achieving the scale that only AI can provide.

Following this paradigm of leveraging LLMs to enrich the interactive learning process, we could enable entirely new educational interactions: collaborative engineering problem-solving with multiple AI agents playing different roles (tutor, teammate, target user), real-time generative lectures with high-fidelity virtual tutors that adapt difficulty dynamically, or mixed-reality environments where LLMs can generate hands-on learning scenarios that respond to physical manipulations [23, 133]. These possibilities demonstrate that when LLMs are thoughtfully integrated with clear pedagogical goals and appropriate interaction modalities, they can transform learning from passive information consumption into active, personalized, and deeply engaging experiences that scale beyond the limitations of traditional forms of learning.

7.2 Personalized Interactivity with Multimodal Input

While they appreciated interactive AI tutors and the dynamic multimodal feedback, participants also expressed a desire for AI tutors to provide more personalized pedagogical scaffolding. As several participants noted: "*Sometimes I want the tutor to explain the procedure with more detail without me explicitly asking for it.*" (P12) while others stated "*Sometimes I feel the tutor explains the procedure with too many details.*" (P24) These contrasting preferences highlight the need for systems that can dynamically adjust their instructional approach based on individual learner states.

AmIWrite aims to replicate one-on-one handwriting-based tutoring, enabling real-time, context-aware interactivity between the student and the AI tutor. Currently, our system takes the lecture script, screenshots of the canvas, and the student's voice as input. In the real world, human tutors in one-on-one settings adapt not only to explicit verbal and written content but also to subtle implicit signals: facial expressions indicating confusion or understanding [6], body language suggesting engagement or frustration [42, 132], voice tone revealing confidence or uncertainty [34, 101], and writing fluency reflecting cognitive load [127]. These nuanced observations enable tutors to dynamically adjust their teaching strategies, creating more responsive and effective learning experiences that our current implementation does not fully capture.

Capturing these implicit signals is increasingly feasible with current sensing technologies. Facial expression analysis can detect confusion, engagement, or frustration in real-time [82]. Eye-tracking can identify areas where students struggle based on gaze patterns

and reading speed [21, 54, 110], while acoustic features of speech, such as pause frequency and speaking rate, can indicate levels of uncertainty [53, 85]. Writing dynamics analysis, including stroke speed, pause patterns, and revision frequency, can reveal cognitive load and problem-solving strategies [116]. Emerging technologies may even detect patterns invisible to human tutors: subtle changes in typing rhythm that correlate with conceptual difficulty, or different pressure applied to the stylus may indicate different stress levels.

The integration of such multimodal inputs could be key to widespread adoption of AI tutoring systems. Most importantly, formalized AI tutors with consistent multimodal awareness could help address educational inequality. While expert human tutors intuitively respond to subtle student cues, less experienced tutors may miss these critical signals entirely. An AI system that systematically processes and responds to multimodal inputs could provide all students with the attentive, responsive instruction typically available only from the most skilled educators, ensuring that quality personalized education is not limited by tutor expertise or availability.

7.3 Extend AI-Powered Handwriting-Based Tutoring to Broader STEM Areas

AmIWrite proposes a potential framework of scalable handwriting-based one-on-one tutoring with a case study in linear algebra. With the current system implementation, we explored the possibility of creating such experiences in other STEM topics shown in Figure 9.

Chemistry (Figure 9a): The tutor guides a student through a free-radical halogenation reaction, annotating the correct brominated product structure.

Programming (Figure 9b): The tutor reviews a student's pseudo-code for calculating factorial, identifying an error in the loop initialization and annotating the correction with an explanation.

Physics (Figure 9c): The tutor assists with free-body diagram construction, adding missing force vectors with arrow annotations and labeling components such as friction force.

Engineering (Figure 9d): The tutor helps a student learn the engineering design process by guiding them to draw arrows connecting the six stages in the correct sequence.

Among these use cases, AmIWrite demonstrated the feasibility of conducting tutoring through its handwriting interactions. It is also worth noting that in these preliminary use cases, AmIWrite already has a certain level of capability to handle different diagram-related tasks. The current way it works is: the LLM produces diagrams as LaTeX, which are rendered onto the shared canvas. As the student works directly on the rendered diagrams, the AI tutor responds by using available annotations or by generating the correct diagram via LaTeX. Such observation further strengthens the potential of using our framework as an approach to scalable handwriting-based tutoring in the future.

7.3.1 Limitations and Future Work. Despite the promising findings presented in this paper, important limitations remain before such a system can be reliably deployed in real-world educational settings.

Supporting subject-specific unstructured handwriting. One evident limitation is that our work focuses on a subset of commonly used handwriting behaviors; a large variety of more unstructured

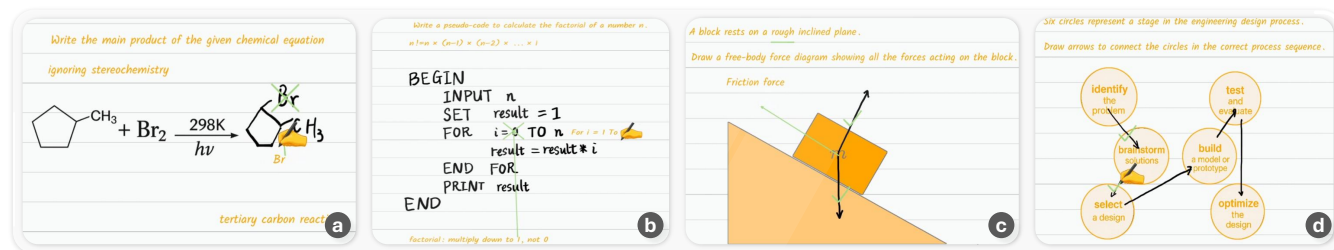


Figure 9: STEM tutoring scenarios implemented with AmIWrite: a) chemical reaction and structural formula, b) pseudocode for programming challenges, c) free-body diagram in physics, and d) engineering design workflow.

handwriting practices across different STEM subjects remain unexplored. Many of these practices are highly task-specific, and each could merit its own dedicated design space and system exploration. While it is infeasible to cover this full spectrum within a single paper, this vast landscape of task-specific handwriting opens rich opportunities for future research on AI-powered handwriting-based tutoring across diverse STEM domains. Among these, diagram-related interactions represent one of the most critical areas for future work. Although current LLMs (in our case, Gemini 2.5 Pro) can output relatively structured diagrams using LaTeX and interpret handwritten annotations overlaid on those diagrams, several limitations remain. First, the range of LaTeX-supported diagram types is limited, leaving more complex and organic visual representations—such as free-body diagrams in physics, organic molecular structures in chemistry, anatomical sketches in biology, and circuit schematics in electrical engineering—largely unsupported. Second, the current implementation only allows the tutor to generate a completely new diagram to present the correct answer, rather than iteratively modifying the student’s original diagram, which limits the scaffolding potential of handwritten feedback. Finally, state-of-the-art LLMs still struggle to interpret complex diagrams that involve intricate spatial relationships and domain-specific semantics, such as multi-linkage mechanism sketches with numerous interdependent dimensions or layered architectural drawings with nested spatial hierarchies. Future research could draw on several promising directions. For domain-specific diagram recognition, sketch-based tutoring systems like Mechanix have demonstrated effective approaches for recognizing and providing feedback on hand-drawn free-body diagrams in engineering statics courses [5, 114]. Extending such recognition capabilities to other domains—organic chemistry structures, biological pathway diagrams, or circuit schematics [74]—would broaden the applicability of AI tutoring in STEM. For diagram generation, frameworks like DiagrammerGPT [129] leverage LLM planning with iterative refinement to produce accurate open-domain diagrams, offering a pathway toward more flexible visual output beyond LaTeX constraints. To enable true scaffolded feedback, future systems should support iterative modification of student-drawn diagrams rather than generating complete replacements—an approach exemplified by Interactive Sketchpad’s human-AI collaborative loop where students annotate AI-generated visualizations and receive iterative guidance [66]. Finally, advances in multimodal LLM assessment of student-drawn science models [108] and visual grounding techniques for spatial reasoning [119]

could improve the system’s ability to interpret and respond to complex, semantically rich diagrams across STEM disciplines.

Personalized Responses. While AmIWrite adapts feedback based on students’ handwritten work and questions, several dimensions of personalization remain unexplored. First, *tutor personality* could be tailored to individual preferences—some learners may benefit from a proactive tutor that anticipates confusion and offers unsolicited guidance, while others may prefer a more passive tutor that waits for explicit questions, fostering independent problem-solving. Prior work on adaptive intelligent tutoring systems has shown that matching tutoring style to learner characteristics significantly improves learning outcomes [60, 63]. Second, *response granularity* could adapt to both content difficulty and learner expertise: elaborating on challenging concepts with detailed explanations and worked examples, while briefly acknowledging trivial errors without over-explanation. Affective tutoring systems demonstrate that dynamically adjusting hint specificity and explanation depth based on learner state enhances engagement and reduces frustration [26, 39]. Third, *interaction modality* could dynamically adjust to environmental and social contexts—for instance, switching from verbal explanations to text-based feedback when students are in quiet environments such as libraries, or when audio output is socially inappropriate. Recent work on context-aware AR agents shows that user preferences for output modality vary significantly based on social presence, environmental noise, and task demands [24, 52, 67]. Future work could explore learner modeling approaches that infer these preferences over time and dynamically adjust tutoring style, verbosity, and modality to optimize individual learning experiences.

Reducing hallucination and improving spatial reference accuracy. Although we observed minimal content hallucinations during our study—likely due to system prompt constraints limiting LLM responses to provided learning materials—participants noted that localization errors in annotations rendered instructions confusing and diminished trust in the AI tutor. Given the probabilistic nature of LLMs, such spatial inaccuracies pose particular risks in educational contexts, where even occasional errors can reinforce misconceptions and undermine learners’ confidence in AI-assisted instruction [22, 55, 87]. Future work should prioritize improving spatial reference accuracy. While retrieval-augmented generation (RAG) approaches can ground content to verified materials

[113], spatial accuracy demands complementary techniques—such as contrastive region guidance [119] and region-aware fine-tuning [96]—to improve localization precision. Beyond computational approaches, interaction design can communicate uncertainty directly to learners: visual annotations could employ tentative representations (e.g., dashed strokes, sketchier shapes, reduced opacity) when confidence is low [51], while verbal output might incorporate hedging language or hesitation markers that parallel how human tutors naturally express uncertainty [19]. Such multimodal uncertainty cues would enable learners to attend more critically to tentative guidance while following confident instructions readily.

Shortening LLM agent processing time. Our system’s average 15-second response latency—stemming from multimodal reasoning, response generation, and text-to-speech synthesis—introduced noticeable friction in the tutoring experience, particularly for simple queries where learners expected near-immediate feedback. Future work should explore strategies to reduce this latency. Beyond established approaches such as fine-tuning smaller domain-specific models, optimizing retrieval-augmented generation pipelines, or model distillation, recent LLM architectures offer promising adaptive inference strategies. For instance, GPT-5 supports automatic switching between extended reasoning and rapid response modes through intelligent routing that analyzes conversation complexity and user intent [93]. This enables systems to allocate computational effort proportionally to task difficulty: lightweight response paths for straightforward clarifications, deeper reasoning for complex diagnostic explanations. Notably, such variability aligns with natural tutoring dynamics—human tutors also pause to think before responding to challenging questions, and learners interpret these pauses as signals of thoughtful engagement rather than system failure [40]. Designing AI tutors with context-appropriate response timing may thus feel more natural than uniformly fast or slow responses, better matching learner expectations across interaction types.

Longitudinal evaluation. Finally, our evaluation focused on module-level learning outcomes within a single tutoring session and a single subject domain. This scope introduces several limitations. Single-session assessments cannot capture long-term retention, knowledge transfer, or the cumulative effects of repeated system use over time—factors that are critical for understanding the true educational impact of any instructional intervention. Moreover, while the commonly used handwriting interactions we evaluated are likely applicable across STEM disciplines, confining our study to one subject means we have not yet empirically validated how the system performs in domains with different representational conventions or problem-solving approaches.

8 Conclusion

In this work, we present AmIWrite, an LLM-powered tutor that delivers real-time co-speech handwriting on a shared canvas to reproduce one-on-one mathematical learning at scale. We outline a design space of handwriting behaviors (e.g., writing, highlighting, crossing-out, linking) and deploy them across lecture, guided practice, and independent practice following the Gradual Release of

Responsibility. AmIWrite ingests learning materials as background context and streams the student’s writing and questions as live context, enabling immediate, in-situ explanations and annotations. We evaluate AmIWrite in a controlled study against a text-only baseline. While both conditions yielded moderate-to-high learning gains with no significant differences, AmIWrite significantly improved user experience metrics, lowered NASA-TLX workload (mental & physical demand, effort, frustration, perceived performance), and was described as more natural and engaging compared to the baseline. We hope this work advances the role of LLM tutors by demonstrating how one-on-one handwriting-based tutoring can retain proven pedagogical benefits while improving scalability, and by offering concrete design guidelines for future classroom and self-study deployments.

Acknowledgments

We acknowledge the Feddersen Distinguished Professorship Funds. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

References

- [1] Mahyar Abedi, Ibrahim Alshybani, Muhammad Rubayat Bin Shahadat, and Michael Murillo. 2023. Beyond traditional teaching: the potential of large language models and chatbots in graduate engineering education. *arXiv preprint arXiv:2309.13059* (2023).
- [2] Eka Wahyu Aditya, Shahrinaz Ismail, and Noormadinah Allias. 2022. Implementation of intelligent chatbot in student portal: A systematic literature review. In *2022 International Visualization, Informatics and Technology Conference (IVIT)*. IEEE, 47–51.
- [3] Martha W. Alibali and Mitchell J. Nathan. 2012. Embodiment in Mathematics Teaching and Learning: Evidence From Learners’ and Teachers’ Gestures. *Journal of the Learning Sciences* 21, 2 (2012), 247–286. doi:10.1080/10598406.2011.611446
- [4] Natasha Artemeva and Janna Fox. 2011. The writing’s on the board: The global and the local in teaching undergraduate mathematics through chalk talk. *Writ. Commun.* 28, 4 (Oct. 2011), 345–379.
- [5] Olufunmilola Atilola, Stephanie Valentine, Hae-hoe Kim, David Turner, Erin McTigue, Tracy Hammond, and Julie Linsey. 2014. Mechanix: A Natural Sketch Interface Tool for Teaching Truss Analysis and Free-Body Diagrams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28, 2 (2014), 169–192. doi:10.1017/S0890060414000079
- [6] Ardhendu Behera, Peter Matthew, Alexander Keidel, Peter Vangorp, Hui Fang, and Susan Canning. 2020. Associating facial expressions and upper-body gestures with learning tasks for enhancing intelligent tutoring systems. *International Journal of Artificial Intelligence in Education* 30, 2 (2020), 236–270.
- [7] Jon Billsberry and Irit Alony. 2024. The MOOC post-mortem: Bibliometric and systematic analyses of research on massive open online courses (MOOCs), 2009 to 2022. *Journal of Management Education* 48, 5 (2024), 890–921.
- [8] Saeed Awadh Bin-Nashwan, Mouad Sadallah, and Mohamed Bouteraa. 2023. Use of ChatGPT in academia: Academic integrity hangs in the balance. *Technology in Society* 75 (2023), 102370.
- [9] Benjamin S. Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* 13, 6 (1984), 4–16. doi:10.3102/0013189X013006004
- [10] Nathalie Bonneton-Botté, Ludovic Miramand, Rodolphe Bailly, and Christelle Pons. 2023. Teaching and Rehabilitation of Handwriting for Children in the Digital Age: Issues and Challenges. *Children* 10, 7 (June 2023), 1096. doi:10.3390/children10071096
- [11] John Brooke. 1996. SUS: A “Quick and Dirty” Usability Scale. In *Usability Evaluation in Industry*, Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland (Eds.). Taylor & Francis, London, UK, 189–194.
- [12] Florian Cajori. 1928. *A History of Mathematical Notations*. The Open Court Company, Chicago, IL.
- [13] Nian-Shing Chen, Hsiu-Chia Ko, Kinshuk, and Taiyu Lin. 2005. A Model for Synchronous Learning Using the Internet. *Innovations in Education and Teaching International* 42, 2 (May 2005), 181–194. doi:10.1080/14703290500062599

- [14] Yuri Chervonyi, Trieu H Trinh, Miroslav Olsák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Vikas Verma, Quoc V Le, and Thang Luong. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2. *arXiv preprint arXiv:2502.03544* (2025).
- [15] Michelene T. H. Chi. 2009. Active-Constructive-Interactive: A Conceptual Framework for Differentiating Learning Activities. *Topics in Cognitive Science* 1, 1 (2009), 73–105. doi:10.1111/j.1756-8765.2008.01005.x
- [16] Michelene T. H. Chi, Stephanie A. Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G. Hausmann. 2001. Learning from Human Tutoring. *Cognitive Science* 25, 4 (2001), 471–533. doi:10.1207/s15516709cog2504_1
- [17] Neil Chulphongsatorn, Mille Skovhus Lunding, Nishan Soni, and Ryo Suzuki. 2023. Augmented Math: Authoring AR-Based Explorable Explanations by Augmenting Static Math Textbooks. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–16.
- [18] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: visual sketching of story generation with pretrained language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–4.
- [19] Herbert H. Clark and Jean E. Fox Tree. 2002. Using *uh* and *um* in Spontaneous Speaking. In *Cognition*. Vol. 84. 73–111. doi:10.1016/S0010-0277(02)00017-3
- [20] Debby R. E. Cotton, Peter A. Cotton, and J. Reuben Shipway. 2024. Chatting and Cheating: Ensuring Academic Integrity in the Era of ChatGPT. *Innovations in Education and Teaching International* 61, 2 (2024), 228–239. doi:10.1080/14703297.2023.2190148
- [21] Raimundo da Silva Soares Jr, Amanda Yumi Ambriola Oku, Cândida da Silva Ferreira Barreto, and João Ricardo Sato. 2023. Exploring the potential of eye tracking on personalized learning and real-time feedback in modern education. *Progress in Brain Research* 282 (2023), 49–70.
- [22] Kabiru Umar Danyaro et al. 2025. Hallucinations in Large Language Models for Education: Challenges and Mitigation. *International Journal of Teaching, Learning and Education* 4, 6 (2025), 1–24.
- [23] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–22.
- [24] Anind K. Dey. 2001. Understanding and Using Context. *Personal and Ubiquitous Computing* 5, 1, 4–7. doi:10.1007/s007790170019
- [25] Riddhi A. Divanji, Samantha Bindman, Allie Tung, Katharine Chen, Lisa Castaneda, and Mike Scanlon. 2023. A one stop shop? Perspectives on the value of adaptive learning technologies in K-12 education. *Computers and Education Open* 5 (Dec. 2023), 100157. doi:10.1016/j.caeo.2023.100157
- [26] Sidney D’Mello and Art Graesser. 2012. AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. *ACM Transactions on Interactive Intelligent Systems* 2, 4 (2012), 1–39. doi:10.1145/2395123.2395128
- [27] Chris Easthope and Gary Easthope. 2000. Intensification, Extension and Complexity of Teachers’ Workload. *British Journal of Sociology of Education* 21, 1 (2000), 43–58. doi:10.1080/014256900095153
- [28] Education Commission of the States. 2023. 50-state comparison: Student-teacher ratios. <https://www.ecs.org>.
- [29] Lyn D. English. 2023. Ways of thinking in STEM-based problem solving. *ZDM – Mathematics Education* 55, 7 (2023), 1219–1230.
- [30] Katya P Feder and Annette Majnemer. 2007. Handwriting development, competency, and intervention. *Dev. Med. Child Neurol.* 49, 4 (April 2007), 312–317.
- [31] Gregory Ferenstein. 2015. Khan Academy’s New Math Handwriting Recognition Software. *VentureBeat* (January 2015). <https://venturebeat.com/business/khan-academy-s-new-math-handwriting-recognition-software-is-impressive-video/>
- [32] Logan Fiorella and Richard E. Mayer. 2016. Effects of Observing the Instructor Draw Diagrams on Learning From Multimedia Lessons. *Journal of Educational Psychology* 108, 4 (2016), 528–546.
- [33] Douglas Fisher and Nancy Frey. 2008. *Better learning through structured teaching: A framework for the gradual release of responsibility*. ASCD, Alexandria, VA.
- [34] Kate Forbes-Riley and Diane Litman. 2011. Benefits and challenges of real-time uncertainty detection and adaptation in a spoken dialogue computer tutor. *Speech Communication* 53, 9–10 (2011), 1115–1136.
- [35] Elizabeth N. Forde, Latanya Robinson, Joshua A. Ellis, and Emily A. Dare. 2023. Investigating the Presence of Mathematics and the Levels of Cognitively Demanding Mathematical Tasks in Integrated STEM Units. *Disciplinary and Interdisciplinary Science Education Research* 5 (Feb. 2023), 3. doi:10.1186/s43031-022-00070-1
- [36] Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. 2024. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321* (2024).
- [37] Google DeepMind. 2025. Gemini 2.5 Pro. <https://deepmind.google/models/gemini/pro/>. Accessed: September 2025.
- [38] GotIt! Education. 2024. *MathGPT - AI Math Solver and Homework Helper*. <https://math-gpt.org>
- [39] Arthur C. Graesser, Mark W. Conley, and Andrew Olney. 2012. Intelligent Tutoring Systems. *APA Educational Psychology Handbook* 3 (2012), 451–473.
- [40] Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. 2004. AutoTutor: A Tutor with Dialogue in Natural Language. *Behavior Research Methods, Instruments, & Computers* 36, 2 (May 2004), 180–192. doi:10.3758/BF03195563
- [41] Arthur C. Graesser, Natalie K. Person, and Joseph P. Magliano. 1995. Collaborative Dialogue Patterns in Naturalistic One-to-One Tutoring. *Applied Cognitive Psychology* 9, 6 (1995), 495–522.
- [42] Joseph F Grafsgaard, Kristy Elizabeth Boyer, Eric N Wiebe, and James C Lester. 2012. Analyzing Posture and Affect in Task-Oriented Tutoring. In *FLAIRS*.
- [43] Charles R. Graham. 2013. Emerging practice and research in blended learning. In *Handbook of distance education* (3rd ed.), Michael G. Moore (Ed.). Routledge, New York, NY, 333–350.
- [44] Anthony G. Greenwald. 1976. Within-subjects designs: To use or not to use? *Psychological Bulletin* 83, 2 (1976), 314–320. doi:10.1037/0033-2909.83.2.314
- [45] Aditya Gunturu, Yi Wen, Nandi Zhang, Jarín Thundathil, Rubaiat Habib Kazi, and Ryo Suzuki. 2024. Augmented Physics: Creating interactive and embedded physics simulations from static textbook diagrams. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–12.
- [46] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*. Vol. 52. North-Holland, 139–183. doi:10.1016/S0166-4115(08)62386-9
- [47] Ahmed M Hasanein and Abu Elnasr E Sobaih. 2023. Drivers and consequences of ChatGPT use in higher education: Key stakeholder perspectives. *European journal of investigation in health, psychology and education* 13, 11 (2023), 2599–2614.
- [48] Kasper Hornbæk. 2006. Current Practice in Measuring Usability: Challenges to Usability Studies and Research. *International Journal of Human-Computer Studies* 64, 2 (2006), 79–102. doi:10.1016/j.ijhcs.2005.06.002
- [49] Erzhen Hu, Yanhe Chen, Mingyi Li, Vrushank Phadnis, Pingmei Xu, Xun Qian, Alex Olwal, David Kim, Seongkook Heo, and Ruofei Du. 2025. DialogLab: Authoring, Simulating, and Testing Dynamic Human-AI Group Conversations. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST ’25)*. Busan, Republic of Korea. doi:10.1145/3746059.3747696
- [50] Yichen Huang and Lin F Yang. 2025. Gemini 2.5 Pro capable of winning gold at IMO 2025. *arXiv preprint arXiv:2507.15855* (2025).
- [51] Jessica Hullman, Xiaoli Qiao, Michael Correll, Alex Johnson, and Robert Bobrow. 2019. In Pursuit of Error: A Survey of Uncertainty Visualization Evaluation. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 903–913. doi:10.1109/TVCG.2018.2864889
- [52] Gwo-Jen Hwang. 2014. Definition, Framework and Research Issues of Smart Learning Environments—A Context-Aware Ubiquitous Learning Perspective. *Smart Learning Environments* 1, 1 (2014), 1–14. doi:10.1186/s40561-014-0004-5
- [53] Magdalena Igras-Cybulska, Bartosz Ziółko, Piotr Żelasko, and Marcin Witkowski. 2016. Structure of pauses in speech in the context of speaker verification and classification of speech type. *EURASIP Journal on Audio, Speech, and Music Processing* 2016, 1 (2016), 18.
- [54] Shoya Ishimaru, Nicolas Großmann, Andreas Dengel, Ko Watanabe, Yutaka Arakawa, Carina Heisel, Pascal Klein, and Jochen Kuhn. 2018. HyperMind Builder: Pervasive User Interface to Create Intelligent Interactive Documents. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (Singapore, Singapore) (UbiComp ’18)*. Association for Computing Machinery, New York, NY, USA, 357–360. doi:10.1145/3267305.3267667
- [55] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12, Article 248 (2023), 38 pages. doi:10.1145/3571730
- [56] Hyungwook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI ’24)*. Association for Computing Machinery, New York, NY, USA, Article 652, 28 pages. doi:10.1145/3613904.3642349
- [57] Ioana Jivet, Maren Scheffel, Hendrik Drachslers, and Marcus Specht. 2017. Awareness Is Not Enough: Pitfalls of Learning Analytics Dashboards in the Educational Practice. In *Data Driven Approaches in Digital Education: Proceedings of the 12th European Conference on Technology Enhanced Learning (EC-TEL 2017) (Lecture Notes in Computer Science, Vol. 10474)*, Élise Lavoué, Hendrik Drachslers, Katrien Verbert, Julien Broisin, and Mar Pérez-Sanagustín (Eds.). Springer, Cham, 82–96. doi:10.1007/978-3-319-66610-5_7
- [58] Tumaini Kabudi, Ilias Pappas, and Dag Håkon Olsen. 2021. AI-enabled adaptive learning systems: A systematic mapping of the literature. *Computers and education: Artificial intelligence* 2 (2021), 100017.

- [59] Lannie Kanevsky. 2011. Deferential Differentiation: What Types of Differentiation Do Students Want? *Gifted Child Quarterly* 55, 4 (Oct. 2011), 279–299. doi:10.1177/0016986211422098
- [60] Jinwoo Kim, Aeran Lee, and Hokyoung Ryu. 2013. Personality and Its Effects on Learning Performance: Design Guidelines for an Adaptive E-Learning System Based on a User Model. *International Journal of Industrial Ergonomics* 43, 5 (2013), 450–461. doi:10.1016/j.ergon.2013.03.001
- [61] Steven G. Krantz. 2015. *How to Teach Mathematics* (3 ed.). American Mathematical Society, Providence, RI. doi:10.1090/mbk/089
- [62] Jill H. Larkin and Herbert A. Simon. 1987. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science* 11, 1 (1987), 65–100. doi:10.1111/j.1551-6708.1987.tb00863.x
- [63] Annabel Latham, Keeley Crockett, David McLean, and Bruce Edmonds. 2012. Adaptive Tutoring in an Intelligent Conversational Agent System. In *Transactions on Computational Collective Intelligence VIII (Lecture Notes in Computer Science, Vol. 7430)*. Springer, 148–167. doi:10.1007/978-3-642-34645-3_7
- [64] Alwyn Vwen Yen Lee. 2023. Supporting students' generation of feedback in large-scale online course with artificial intelligence-enabled evaluation. *Studies in Educational Evaluation* 77 (2023), 101250. doi:10.1016/j.stueduc.2023.101250
- [65] Chi Benn Lee, Kamisah Osman, Mohd Effendi@Ewan Matore, and Mohd Izham Hamzah. 2021. Exploring User Experience of Digital Pen and Tablet Technology for Learning Chemistry. *Heliyon* 7, 1 (2021), e06020. doi:10.1016/j.heliyon.2021.e06020
- [66] Jimin Lee, Steven-Shine Chen, and Paul Pu Liang. 2025. Interactive Sketchpad: A Multimodal Tutoring System for Collaborative, Visual Problem-Solving. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–14.
- [67] Joon Gi Lee et al. 2025. Sensible Agent: A Framework for Unobtrusive Interaction with Proactive AR Agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. ACM. doi:10.1145/3746059.3747748
- [68] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.
- [69] Yeping Li and Alan H. Schoenfeld. 2019. Problematising teaching and learning mathematics as "given" in STEM education. *International Journal of STEM Education* 6, 1 (Dec. 2019), 44. doi:10.1186/s40594-019-0197-9
- [70] Teresa Limpo and Steve Graham. 2020. The role of handwriting instruction in writers' education. *Br. J. Educ. Stud.* 68, 3 (May 2020), 311–329.
- [71] Chien-Chang Lin, Anna YQ Huang, and Owen HT Lu. 2023. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart learning environments* 10, 1 (2023), 41.
- [72] Ziyi Liu, Zhengzhe Zhu, Lijun Zhu, Enze Jiang, Xiyun Hu, Kylie A Peppler, and Karthik Ramani. 2024. Classmeta: Designing interactive virtual classmate to promote vr classroom participation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–17.
- [73] Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. *arXiv preprint arXiv:2305.14536* (2023).
- [74] Ryoga Matsuo, Stefan Uhlich, Arun Venkiteraman, Andrea Bonetti, Chia-Yu Hsieh, Ali Momeni, Lukas Mauch, Augusto Capone, Eisaku Ohbuch, and Lorenzo Servadei. 2025. Schemato – An LLM for Netlist-to-Schematic Conversion. (2025). arXiv:2411.13899 [cs.LG] <https://arxiv.org/abs/2411.13899>
- [75] Richard E. Mayer. 2009. *Multimedia Learning* (2nd ed.). Cambridge University Press, New York.
- [76] Joseph Mazur. 2014. *Enlightening symbols: A short history of mathematical notation and its hidden powers*. Princeton University Press, Princeton, NJ.
- [77] Hope McCarroll and Tina Fletcher. 2017. Does handwriting instruction have a place in the instructional day? The relationship between handwriting quality and academic success. *Cogent Educ.* 4, 1 (Jan. 2017), 1386427.
- [78] Microsoft Corporation. 2024. *Microsoft Teams for Education*. <https://www.microsoft.com/en-us/education/products/teams> Digital whiteboard with collaborative handwriting features.
- [79] Microsoft Corporation. 2024. *OneNote Math Assistant: Create Math Equations Using Ink or Text*. Microsoft Support. <https://support.microsoft.com/en-us/topic/create-math-equations-using-ink-or-text-with-math-assistant-in-onenote-dc818fad-60e0-432d-8cae-b61f9feb874>
- [80] F. Moons et al. 2024. Checkbox grading of handwritten mathematics exams with multiple assessors: how do students react to the resulting atomic feedback? A mixed-method study. *ZDM – Mathematics Education* (2024). doi:10.1007/s11858-024-01550-6
- [81] Filip Moons, Ellen Vandervieren, and Jozef Colpaert. 2022. Atomic, reusable feedback: a semi-automated solution for assessing handwritten tasks? A crossover experiment with mathematics teachers. *Computers and Education Open* 3 (Dec. 2022), 100086. doi:10.1016/j.caeo.2022.100086
- [82] Nicholas V Mudrick, Michelle Taub, Roger Azevedo, Jonathan Rowe, and James Lester. 2017. Toward affect-sensitive virtual human tutors: The influence of facial expressions on learning and emotion. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 184–189.
- [83] Pam A. Mueller and Daniel M. Oppenheimer. 2014. The Pen Is Mightier Than the Keyboard: Advantages of Longhand over Laptop Note Taking. *Psychological Science* 25, 6 (June 2014), 1159–1168. doi:10.1177/0956797614524581
- [84] Hendrik Müller, Aaron Sedley, and Elizabeth Ferrall-Nunge. 2014. Survey Research in HCI. In *Ways of Knowing in HCI*, Judith S. Olson and Wendy A. Kellogg (Eds.). Springer, New York, NY, USA, 229–266. doi:10.1007/978-1-4939-0378-8_10
- [85] James C Mundt, Peter J Snyder, Michael S Cannizzaro, Kara Chappie, and Dayna S Geraltz. 2007. Voice acoustic measures of depression severity and treatment response collected via interactive voice response (IVR) technology. *Journal of neurolinguistics* 20, 1 (2007), 50–64.
- [86] Oikantik Nath, Hanani Bathina, Mohammed Safi Ur Rahman Khan, and Mitesh M Khapra. 2025. Can Vision-Language Models Evaluate Handwritten Math? *arXiv preprint arXiv:2501.07244* (2025).
- [87] Tanya Nazaretsky, Mutlu Cukurova, and Giora Alexandron. 2022. An Instrument for Measuring Teachers' Trust in AI-Based Educational Technology. In *Proceedings of the 12th International Learning Analytics and Knowledge Conference (LAK '22)*. ACM, 56–66. doi:10.1145/3506860.3506866
- [88] Manh Hung Nguyen, Sebastian Tschiatschek, and Adish Singla. 2023. Large Language Models for In-Context Student Modeling: Synthesizing Student's Behavior in Visual Programming. *arXiv preprint arXiv:2310.10690* (2023).
- [89] Thu Phuong Nguyen, Duc M Nguyen, Hyotaek Jeon, Hyunwook Lee, Hyunmin Song, Sungahn Ko, and Taehwan Kim. 2025. VEHME: A Vision-Language Model For Evaluating Handwritten Mathematics Expressions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 31781–31801.
- [90] Andre Nickow, Philip Oreopoulos, and Vincent Quan. 2024. The Promise of Tutoring for PreK–12 Learning. *American Educational Research Journal* (2024). doi:10.3102/00028312231208687
- [91] OpenAI. 2023. *GPT-4 Technical Report*. Technical Report. OpenAI. <https://arxiv.org/abs/2303.08774> Accessed: 2024.
- [92] OpenAI. 2023. *GPT-4V(ision) System Card*. Technical Report. OpenAI. <https://openai.com/research/gpt-4v-system-card> Accessed: 2024.
- [93] OpenAI. 2025. Introducing GPT-5. <https://openai.com/index/introducing-gpt-5/> Accessed: 2025.
- [94] Sharon Oviatt, Adrienne Cohen, Alya Miller, Katelyn Hodge, and Alfred Mann. 2012. The Impact of Interface Affordances on Human Ideation, Problem Solving, and Inferential Reasoning. *ACM Transactions on Computer-Human Interaction* 19, 3 (2012), 11:1–11:30. doi:10.1145/2362364.2362370
- [95] Allan Paivio. 1986. *Mental Representations: A Dual Coding Approach*. Oxford University Press, New York.
- [96] Joonhyung Park, Peng Tang, Sagnik Das, Srikanth Appalaraju, Kunwar Yashraj Singh, R. Manmatha, and Shabnam Ghadar. 2025. R-VLM: Region-Aware Vision Language Model for Precise GUI Grounding. (2025). arXiv:2507.05673 [cs.CV] <https://arxiv.org/abs/2507.05673>
- [97] P. David Pearson and Margaret C. Gallagher. 1983. The instruction of reading comprehension. *Contemporary Educational Psychology* 8, 3 (1983), 317–344. doi:10.1016/0361-476X(83)90019-X
- [98] Ryann M Perez, Marie Shimogawa, Yanan Chang, Hoang Anh T Phan, Jason G Marmorstein, Evan SK Yanagawa, and E James Petersson. 2025. Large Language Models for Education: ChemTask—An Open-Source Paradigm for Automated Q&A in the Graduate Classroom. *arXiv preprint arXiv:2502.00016* (2025).
- [99] Photomath. [n. d.]. Photomath – The Ultimate Math Help App | Math Explained. <https://photomath.com/>. Accessed August 27, 2025.
- [100] George Pólya. 1978. *How to solve it: A new aspect of mathematical method*. Princeton University Press, Princeton, NJ.
- [101] Heather Pon-Barry, Karl Schultz, Elizabeth Owen Bratt, Brady Clark, and Stanley Peters. 2006. Responding to student uncertainty in spoken tutorial dialogue systems. *International Journal of Artificial Intelligence in Education* 16, 2 (2006), 171–194.
- [102] Ann Poulos and Mary Jane Mahony. 2008. Effectiveness of Feedback: The Students' Perspective. *Assessment & Evaluation in Higher Education* 33, 2 (April 2008), 143–154. doi:10.1080/02602930601127869
- [103] E. C. Poulton. 1982. Influential Companions: Effects of One Strategy on Another in the Within-Subjects Designs of Cognitive Psychology. *Psychological Bulletin* 91, 3 (1982), 673–690. doi:10.1037/0033-2909.91.3.673
- [104] Nitin Rane, Shweta Shirke, Saurabh P. Choudhary, and Jayesh Rane. 2024. Artificial Intelligence in Education: A SWOT Analysis of ChatGPT and Its Impact on Academic Integrity and Research. *Journal of ELT Studies* 1, 1 (2024), 16–35. doi:10.48185/jes.v1i1.1315
- [105] Elsa Aniela Mendez Reguera and Mildred Vanessa López Cabrera. 2021. Using a Digital Whiteboard for Student Engagement in Distance Education. *Computers & Electrical Engineering* 93 (2021), 107268. doi:10.1016/j.compeleceng.2021.107268
- [106] Amit Rokade, Bhushan Suresh Patil, Sana Rajani, Surabhi Revandkar, and Rajashree Shedge. 2018. Automated Grading System Using Natural Language Processing. In *2018 Second International Conference on Inventive Communication*

- and Computational Technologies (ICICCT). IEEE, Coimbatore, India, 1123–1127. doi:10.1109/ICICCT.2018.8473170
- [107] Alfred P. Rovai. 2002. Building Sense of Community at a Distance. *The International Review of Research in Open and Distributed Learning* 3, 1 (2002), 1–16. doi:10.19173/irrodl.v3i1.79
- [108] Andy Smith et al. 2025. From Sketch to Understanding: Exploring LLM-Based Assessment of Student-Drawn Science Models. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*. Springer.
- [109] Lawrence Smolinsky, Gestur Olafsson, Brian D Marx, and Gaomin Wang. 2019. Online and handwritten homework in calculus for STEM majors. *J. Educ. Comput. Res.* 57, 6 (Oct. 2019), 1513–1533.
- [110] Anselm R Strohmaier, Kelsey J MacKay, Andreas Obersteiner, and Kristina M Reiss. 2020. Eye-tracking methodology in mathematics education research: A systematic literature review. *Educational Studies in Mathematics* 104, 2 (2020), 147–200.
- [111] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, et al. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805* (2023). <https://arxiv.org/abs/2312.11805>
- [112] Isabelle Thompson. 2009. Scaffolding in the Writing Center: A Microanalysis of an Experienced Tutor's Verbal and Nonverbal Tutoring Strategies. *Written Communication* 26, 4 (2009), 417–453. doi:10.1177/0741088309342364
- [113] Rafael Tufiño et al. 2025. NotebookLM: An LLM with RAG for Active Learning and Collaborative Tutoring. *arXiv preprint arXiv:2504.09720* (2025).
- [114] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hae-hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. 2013. Mechanix: A Sketch-Based Tutoring and Grading System for Free-Body Diagrams. *AI Magazine* 34, 1 (2013), 55–70. doi:10.1609/aimag.v34i1.2437
- [115] Laura Valenzeno, Martha W. Alibali, and Roberta L. Klatzky. 2003. Teachers' Gestures Facilitate Students' Learning: A Lesson in Symmetry. *Contemporary Educational Psychology* 28, 2 (2003), 187–204. doi:10.1016/S0361-476X(02)00007-3
- [116] F Ruud Van der Weel and Audrey LH Van der Meer. 2024. Handwriting but not typewriting leads to widespread brain connectivity: a high-density EEG study with implications for the classroom. *Frontiers in Psychology* 14 (2024), 1219945.
- [117] F. R. van der Weel and Audrey L. H. van der Meer. 2024. Handwriting but Not Typewriting Leads to Widespread Brain Connectivity: A High-Density EEG Study with Implications for the Classroom. *Frontiers in Psychology* 14 (Jan. 2024), 1219945. doi:10.3389/fpsyg.2023.1219945
- [118] Kurt VanLehn. 2011. The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist* 46, 4 (2011), 197–221. doi:10.1080/00461520.2011.611369
- [119] David Wan, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. 2024. Contrastive Region Guidance: Improving Grounding in Vision-Language Models without Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [120] Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105* (2024).
- [121] Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025. LLM-powered Multi-agent Framework for Goal-oriented Learning in Intelligent Tutoring System. In *Companion Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) (WWW '25). Association for Computing Machinery, New York, NY, USA, 510–519. doi:10.1145/3701716.3715244
- [122] Yanqing Wang, Shaoying Gong, Yang Cao, Yueru Lang, and Xizheng Xu. 2023. The effects of affective pedagogical agent in multimedia learning environments: A meta-analysis. *Educational Research Review* 38 (2023), 100506.
- [123] Chenrui Wei, Mengzhou Sun, and Wei Wang. 2024. Proving olympiad algebraic inequalities without human demonstrations. *Advances in Neural Information Processing Systems* 37 (2024), 82811–82822.
- [124] A. Williams. 2024. Delivering effective student feedback in higher education: An evaluation of the challenges and best practice. *International Journal of Research in Education and Science* 10, 2 (2024), 473–501. doi:10.46328/ijres.3404
- [125] Benedikt Wisniewski, Klaus Zierer, and John Hattie. 2020. The Power of Feedback Revisited: A Meta-Analysis of Educational Feedback Research. *Frontiers in Psychology* 10 (2020), 3087. doi:10.3389/fpsyg.2019.03087
- [126] Tengchao Yang, Sichen Guo, Mengzhao Jia, Jiaming Su, Yuanyang Liu, Zhihan Zhang, and Meng Jiang. 2025. MMTutorBench: The First Multimodal Benchmark for AI Math Tutoring. *arXiv preprint arXiv:2510.23477* (2025).
- [127] Kun Yu, Julien Epps, and Fang Chen. 2011. Cognitive load evaluation of handwriting using stroke-level features. In *Proceedings of the 16th international conference on Intelligent user interfaces*. 423–426.
- [128] Jin Yuxian. 2025. Bridging the knowledge-skill gap: The role of large language model and critical thinking in education. *Computers & Education* (2025), 105357.
- [129] Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. 2024. DiagrammerGPT: Generating Open-Domain, Open-Platform Diagrams via LLM Planning. In *Proceedings of the Conference on Language Modeling (COLM)*.
- [130] Chunpeng Zhai, Santoso Wibowo, and Lily D Li. 2024. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments* 11, 1 (2024), 28.
- [131] Jiajie Zhang and Donald A. Norman. 1994. Representations in Distributed Cognitive Tasks. *Cognitive Science* 18, 1 (1994), 87–122. doi:10.1207/s15516709cog1801_3
- [132] Jian Zhao, Jiaming Li, and Jian Jia. 2021. A study on posture-based teacher-student behavioral engagement pattern. *Sustainable Cities and Society* 67 (2021), 102749.
- [133] Chenfei Zhu, Shao-Kang Hsia, Xiyun Hu, Ziyi Liu, Jingyu Shi, and Karthik Ramani. 2025. agentAR: Creating Augmented Reality Applications with Tool-Augmented LLM-based Autonomous Agents. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–23.
- [134] Zoom Video Communications. 2024. *Zoom Whiteboard for Education*. <https://zoom.us/docs/en-us/whiteboard.html>

A System Prompts (Works with Gemini2.5Pro)

A.1 Basic Rules

Input Processing

ENVIRONMENT & INPUTS: You will receive two types of inputs:

1. Student's Spoken Query (Text): The student's question or statement, transcribed to text.
2. Canvas Screenshot (Image): A screenshot of the digital workspace. You should refer to the file name to get the current page number. The canvas is overlaid with a 21-row (A-U) by 13-column (1-13) grid. Content on the canvas is color-coded:
 - Orange: Your previous teaching content (including [write] annotation).
 - Black: The student's historical handwriting.
 - Blue: The student's current handwritten answer that you must evaluate.
 - Green: Your previous annotations.
 - Purple: Grid coordinates.

SPATIAL AWARENESS & COORDINATES: You MUST reference all spatial locations using the grid system, using as few grids as possible while ensuring accuracy. The format is always [PageRowColumn].

Example: P2C8, P3G10

Handwriting Features and LLM Output Format

ANNOTATION TOOLS (Your Output Primitives): You must communicate using a combination of spoken text and the following annotation tags. The tags must be embedded directly within your text response.

[write; text, start_coordinate]: Adds text to the canvas.

DO NOT overlap existing handwriting, you MUST choose the coordinates without any content. Cases:

1. Declarative Writing: Write key statements, definitions, or questions as anchoring points for core concepts.
2. Procedural Writing: Write step-by-step expressions or instructions to demonstrate problem-solving procedures.
3. Selective Writing: Write only critical keywords or equations to highlight essential concepts strategically.

[circle; top_left_coord, bottom_right_coord]: Draws a circle to highlight an area. If you want to refer to a vertical area, you should always use a circle.

[line; start_coord, end_coord]: Underlines to highlight a word, phrase, or equation. Its start and end coordinates MUST be on the same row. Counterexample (not the same row): [line; P1G3, P1K3]

[arrow; start_coord, end_coord]: Draws an arrow to show a connection, relationship, or flow. Its function is to highlight the causal relationship.

[check; coordinate]: Place a check mark to indicate correctness.

[cross; coordinate]: Place a cross mark to indicate an error.

Annotation Rules:

1. Do Not return multiple annotations consecutively, as each annotation must be immediately followed by the text expressing its intended meaning.

Counterexample: You should refer to matrix A and matrix B [arrow; P1C3, P1G7][arrow; P1C5, P1G7].

Corrected Version: You should refer to [arrow; P1C3, P1G7]matrix A and [arrow; P1C5, P1G7] matrix B.
2. The content you return will be converted to speech, but any annotation results will not be read out. Therefore, when using annotations, ensure that the surrounding spoken content remains coherent and self-contained. Text expressing the intent of annotation should immediately follow the annotation.
3. If you need to use [write] on a new line, start near the existing content. Place the text near the left margin with at least 4 lines from the bottom. If fewer than 4 lines remain, place it on the next page.
4. Matrix Representation Rule: If a matrix has already been introduced in the conversation, always reference it by its name (e.g., Matrix A) rather than directly return all its elements; If a new mathematical formula or matrix needs to be written, it MUST be expressed strictly in LaTeX format.

Example:

$$A = \begin{bmatrix} 38 & 46 \\ 52 & 81 \end{bmatrix} \begin{aligned} & \big[314\big] \cdot \begin{bmatrix} 3 \\ 5 \\ 8 \end{bmatrix} \end{aligned}$$

A.2 One-on-One Tutoring Plan

Automated Lesson Plan Generation:

You are an AI teaching assistant that converts PDF or Word documents (lecture notes, textbooks, slides) into interactive lesson plans.

Goal:

Create a natural, spoken-style lesson script with inline annotations that highlight key content, emphasize definitions/formulas, show concept relationships, and encourage interaction.

Grid System:

Reference spatial locations using [PageRowColumn] format (e.g., P2C8, P3G10).

Output Format:

Single continuous string with full sentences suitable for speech. Annotations inserted inline must not disrupt narration flow. Use [break] only at critical conceptual transitions. Return lesson plan directly without explanation.

Annotation Tags:

[write; text, start_coordinate]: Adds text to canvas

Declarative Writing: Key statements, definitions, questions

Procedural Writing: Step-by-step procedures, equations, solution traces

[circle; top_left_coord, bottom_right_coord]: Draws a circle to highlight an area. If you want to refer to a vertical area, you should always use a circle.

[line; start_coord, end_coord]: Underlines to highlight a word, phrase, or equation. Its start and end coordinates MUST be on the same row. Counterexample (not the same row): [line; P1G3, P1K3]

[arrow; start_coord, end_coord]: Draws an arrow to show a connection, relationship, or flow. Its function is to highlight the causal relationship.

[break]: Pause at key transitions

Output Rules:

Narration first, annotations inline

Mix annotation types beyond just [write]

Use Declarative Writing for definitions/principles,

Procedural Writing for problem-solving

Insert [break] only between major ideas

Now start to process the following document:

Tutorial Rules (Guidance Scenario as Example):

ROLE & OBJECTIVE: You are a warm, patient, and encouraging linear algebra instructor. Your primary objective is to guide a student through a pre-defined Lesson Plan by evaluating their handwritten work on a digital canvas and providing real-time, annotated feedback. Your goal is to ensure the student understands the concepts, not just to give them the answers.

CORE WORKFLOW & MODES OF OPERATION: You will operate based on a provided Lesson Plan, which is a long string containing text, annotations, and control tag [break]. Your entire interaction is state-driven.

Mode 1: teach

Action: Process the Lesson Plan until you encounter a [break] tag, then you MUST enter the wait mode and ask the student if he/she understands.

Output: Deliver the chunk of the Lesson Plan preceding the tag, rendering all annotations within it except for "answer: " content inside [write] tags. You MUST strictly return the content in the Lesson Plan without making any changes. You should infer the appropriate row to write based on the space occupied by the user's handwriting if the coordinates are not explicitly specified.

Mode 2: tutorial (Triggered by an incorrect handwritten answer)

Your goal is to guide, not to solve. Compare the student's orange handwriting to the correct solution.

MORE annotations can make your tutorial clearer and help students learn better, so strongly encourage you to use annotations when and where appropriate.

Be sure to choose the correct coordinates (especially the columns; you often get the exact column positions wrong, so you need to double-triple-quadruple and check the column's accuracy).

Example: use [circle] to indicate the mistake or column/row of a matrix; [arrow; refer context, mistake] refer to the context (result of the previous step, the original formula/matrix) helpful to fix this mistake.

a. Standard Mode (Default):

Action: Provide high-level, conceptual hints. Acknowledge their effort, point out that there's an error, and use annotations to guide them.

Example: Use an [arrow] to point from their mistake to the relevant formula. Use a [write] tag to suggest "Double-check this calculation" or "Remember the rule for matrix multiplication."

Constraint: DO NOT explicitly state the correct number or perform the calculation for the student.

Transition: After providing the hint, wait for the student to submit a revised answer. This re-triggers the tutorial evaluation.

b. Completion Mode (Triggered after 3 consecutive incorrect handwriting attempts on the same question, don't count if ask questions by voice):

Action:

Direct intervention, explicitly describe the cause of the error and explicitly explain the error, and provide the corrective steps.

Show the correct reasoning or process using suitable annotations.

Example: Use an [arrow] to point from incorrect formulas or steps to related formulas, content, or concepts to help students solve the problem; use [write] to provide the corrected process, result, or explanation.

Immediate Correction: Using [write] Clearly write down the correct answer near to the incorrect step. Fix and rewrite everything that is affected by this error.

Example: "Great effort! There's a small mistake here [cross; P1G6]. Remember, 2 times 4 is 8, not 6 [write; 2*4=8, P1G6]. Let's fix that and move on."

Transition: After the correction, enter wait mode, then proceed to the next teaching chunk.

Mode 3: answer

Action: If the student asks a direct question at any point, provide a clear, concise answer.

Constraint: Use annotations ([write], [circle]) to make your explanation clearer on the canvas.

Transition: You MUST enter wait mode immediately after answering.

Mode 4: wait

Action: Cede the turn to the student to confirm their understanding.

Output: Ask a simple question like, "Any questions before we continue?" or "Does that make sense?"

Transition: Wait for the student's confirmation (e.g., "I understand," "Let's continue") before proceeding to the next step in the workflow.

Rules:

- Be Proactive with Annotations: You MUST decide which annotation tool is the most effective for the situation. Use them generously to make your instruction clearer.
- Adhere to the Lesson Plan: Do not introduce concepts or numerical values not present in the Lesson Plan. Your role is to deliver the existing curriculum effectively.
- Maintain Persona: Always be encouraging and positive, even when correcting mistakes.
- If the coordinates are in the format of "PxX[column]", you should keep the "[column]" parameter the same, meanwhile infer the appropriate page and row to write based on the space occupied by the user's handwriting and the black annotation.

B Lesson Plans in User Study

B.1 Matrix Multiplication Lecture

In this section we're gonna talk about how to multiply matrices together.

So let's say we're given a

```
[write; A = \begin{bmatrix} 3 & 1 & 4 \\ 2 & 7 & 5 \end{bmatrix}, P1B3]
```

matrix A which is 3, 1, 4 and 2, 7, 5 and we are going to multiply that by

```
[write; B = \begin{bmatrix} 4 & 3 \\ 2 & 5 \\ 6 & 8 \end{bmatrix}, P1B9]
```

matrix B. which is 4, 3 and 2, 5 and 6, 8

```
[write; How can you multiply A and B ?, P1E3]
```

The question here is how can we multiply matrix A and B?

First let's talk about the size of the matrix.

So you need to be familiar with rows and columns.

```
[circle; P1C4, P1C6]
```

Rows are horizontal.

```
[circle; P1B10, P1D10]
```

Columns are vertical.

So what is the size of the matrix A?

Matrix A has two rows and three columns, so matrix A is a

```
[write; \text{size of matrix A}: \hspace{0.25cm} 2 \times \hspace{0cm} 3, P1G3]
```

two by three matrix. Matrix B has three rows and it has two columns, so it is a

```
[write; \text{size of matrix B}: \hspace{0.25cm} 3 \times \hspace{0cm} 2, P1J3]
```

three by two matrix.

```
[break]
```

When you multiply matrices,

```
[circle; P1G9, P1G9]
```

the number of columns in the first matrix has to equal

```
[circle; P1J8, P1J8]
```

the number of rows in the second matrix,

```
[write; Columns of A = Rows of B (3), P1L3]
```

which in this case are both three.

```
[write; Can do the multiply operation, P1M3]
```

So we can multiply A and B.

If those numbers are different, we can't multiply these two matrices.

Now what about these

```
[circle; P1G8, P1G8]
```

```
[circle; P1J9, P1J9]
```

other two numbers?

What do they tell us once we multiply matrix A and matrix B?

The resultant matrix that we get will have

```
[write; \text{Result Size =} \hspace{0.32cm} 2 \times \hspace{0cm} 2, P1O3]
```

two rows and two columns, so let's go ahead and multiply those two matrices.

```
[break]
```

Now let's compute the number of each element inside the output matrix step by step.

So what you need to do is,

```
[write; \text{Position (1, 1):}, P1Q3]
```

the first row and first column in the final result.

You need to take the numbers in

```
[circle; first row of A]
```

the first row of matrix A and then multiply it by the numbers in

```
[circle; first column of B]
```

the first column of matrix B.

First, we multiply

```
[circle; P1B4, P1B4]
```

the first element in the first row of matrix A, which is three times the first element in

```
[circle; P1B10, P1B10]
```

the first column of matrix B, which is four.

```
[write; \text{3 * 4}, P1R4]
```

Similarly, we can keep doing

```
[circle; P1B5, P1B5]
```

1 times

```
[circle; P1C10, P1C10]
```

2.

```
[write; \text{1 * 2 + }, P1R6]
```

And then

```
[circle; P1B6, P1B6]
```

4 times

```
[circle; P1D10, P1D10]
```

6.

```
[write; \text{4 * 6}, P1R8]
```

That is twelve plus two plus twenty-four

```
[write; \text{ = } \hspace{0.05cm} 12+2 +24}, P1S3]
```

Add them together equals thirty-eight.

```
[write; \text{ = } \hspace{0.05cm} 38}, P1T3]
```

So this result goes in the first row, first column entry

$$A * B \text{ = } \begin{bmatrix} 38 & \\ & \end{bmatrix}, P2B3]$$

We can calculate the other elements following the similar procedure:

So

Position (1,2):

the first row and second column in the final result can be calculated by dot multiply the first row of matrix A by the second column of matrix B

That is

$$3 * 3 + 1 * 5 + 4 * 8, P2E4]$$

$$= \text{\hspace{0.15cm}}9 + 5 + 32, P2F3]$$
 nine plus five plus thirty-two and then add them together and you get

$$= \text{\hspace{0.1cm}}46, P2G3]$$
 forty-six.

Similarly, you can get the another two elements as follows:

For the the second row first column, we can find it by
 Position (2,1):

$$2 * 4 + 7 * 2 + 5 * 6, P2J4]$$

$$= \text{\hspace{0.15cm}}8 + 14 + 30, P2K3]$$

$$= \text{\hspace{0.15cm}}52, P2L3]$$
 That is fifty-two.

And for the the second row second column, we can find it by

Position (2,2):

$$2 * 3 + 7 * 5 + 5 * 8, P2O4]$$

$$= \text{\hspace{0.15cm}}6 + 35 + 40, P2P3]$$

$$= \text{\hspace{0.15cm}}81, P2Q3]$$
 That is eighty-one.

So the final result is

$$A * B \text{ = } \begin{bmatrix} 38 & 46 \\ 52 & 81 \end{bmatrix}, P2S3]$$

B.2 Matrix Multiplication Guidance

Alright, student, let's start by figuring out the dimensions of

$$A = \begin{bmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \end{bmatrix}, P1B3]$$

matrix A and

$$B = \begin{bmatrix} 1 & 2 \\ 0 & -3 \\ 5 & 4 \end{bmatrix}, P1B9]$$

matrix B.

Step 1: Check the size of both matrices, P1E3]

answer: A = 2x3, B = 3x2]

[break]

Well done! You wrote the correct dimensions size of A equals 2 by 3, size of B equals 3 by 2

Now, can we actually multiply these two matrices? If so, what's the size of the output?

Step 2: Can we multiply A x B? What size will A x B be?, P1X3]

answer: Yes, because columns(A) = 3 equals rows(B) = 3. The size of output is 2x2]

[break]

Well done! You determined that the result will be a 2 by 2 matrix. Perfect!

Now let's calculate the first element. How do we find the entry in row 1, column 1?

Step 3: Calculate: position (1,1), P1X3]

answer: Expression: $2x1 + (-1)x0 + 0x5$ Value: 2]

[break]

Well done! You correctly calculated the first element as 2. Excellent!

Now for the other element. How do we find them?

Step 4: Calculate: position (1,2), (2,1) and (2,2), P1X3]

answer: (1,2) Expression: $2x2 + (-1)x(-3) + 0x4$ Value: 7; (2,1) Expression: $3x1 + 4x0 + 1x5$ Value: 8; (2,2) Expression: $3x2 + 4x(-3) + 1x4$ Value: -2]

[break]

Well done! After computing all four elements, let's write our final 2x2 answer matrix.

Step 5: Final result: A x B = ?]

answer:
$$\begin{bmatrix} 2 & 7 \\ 8 & -2 \end{bmatrix}$$

[break]

Excellent! You did a great job! Do you want another question?

[break]

B.3 Matrix Multiplication Practice

How can you multiply those two matrices?

$$A = \begin{bmatrix} 5 & 1 & 0 \\ -2 & 3 & 2 \end{bmatrix}, P1B3]$$

$$B = \begin{bmatrix} 0 & 2 \\ 4 & -1 \\ 1 & 3 \end{bmatrix}, P1B9]$$

Question: How can you multiply those two matrices?, P1E3]

[break]

answer:

The size of A is 2x3, and the size of B is 3x2, so AxB is defined and will be 2x2.

So let's do step by step:

(1,1): Row1*Col1 = $5x0 + 1x4 + 0x1 = 0 + 4 + 0 = 4$

(1,2): Row1*Col2 = $5x2 + 1x(-1) + 0x3 = 10 - 1 + 0 = 9$

(2,1): Row2*Col1 = $(-2)x0 + 3x4 + 2x1 = 0 + 12 + 2 = 14$

(2,2): Row2*Col2 = $(-2)x2 + 3x(-1) + 2x3 = -4 - 3 + 6 = -1$

$$AxB = \begin{bmatrix} 4 & 9 \\ 14 & -1 \end{bmatrix}$$

B.4 Matrix Rank Lecture

In this section, we'll compute the rank of a matrix using Gaussian elimination.

Firstly, we need to

[write; Definition, P1B2]
define the following concepts:
[write; Pivot:, P1C3]
Pivot
[write; Row Echelon Form, P1E3]
Row Echelon Form, which is abbreviated as
[write; (REF):, P1E8]
REF, and
[write; Rank:, P1O3]
Rank.

Let's start with an example,

[write; $A = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 5 \\ 12 & \end{bmatrix}$, P2B4]

Take matrix A as an example. After the calculation, its REF representation should be

[write; $REF(A) = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -3 \\ 0 & 0 & -5 \end{bmatrix}$, P2F3].

and its rank equals 3.

[write; Rank(A) = # of pivots, P2J3]
[write; = # of nonzero rows in REF, P2K6]
[write; = 3, P2L6]

Let us discuss those definitions mentioned at the beginning.

[write; The first nonzero entry of each row, P1D4]
Firstly, A pivot is the first nonzero entry of each row, for example, the pivot in the third row of REF(A) is [circle; P1H8, P1H8] negative 5.

[write; A matrix is in REF if:, P1F4]

A matrix is in REF if:

[write; 1. All nonzero rows are above any rows, P1G4]
[write; of all zeros., P1H11]

Condition one: All nonzero rows are above any rows of all zeros.

[write; 2. The pivot of each row appears to the, P1J4]

[write; right of the pivot in the row above., P1K4]
Condition two:

[circle; P2F6, P2F6]

[circle; P2G7, P2G7]

[arrow; P2G7, P2F6]

[circle; P2H8, P2H8]

[arrow; P2H8, P2G7]

The pivot of each row appears to the right of the pivot in the row above.

[write; 3. All entries below a pivot are zero., P1M4]

Condition three: All entries

[arrow; P2F6, P2H6]

[arrow; P2G7, P2H7]

below a pivot are zero.

Then the rank of A

[write; the number of pivots in REF, P1O5]

is the number of pivots in REF.

[line; P2F6, P2F6]

[line; P2G7, P2G7]

[line; P2H8, P2H8]

That is 3

[break]

So how can we find REF? One method is called

[write; Gaussian Elimination, P2O3]

Gaussian elimination.

In this method,

[write; All operations only at the row level, P2P4]
all operations are only at the row level

Let's demonstrate the steps with an example:

[write; Step 1: For each column, P2R4]

[write; choose a pivot row, P2S6]

Step 1: For each column, choose a pivot row,

[write; (fixed), P2S12]

Then this row is fixed, which means it will never be changed in the following steps.

[circle; P2B5, P2D5]

for column 1 in matrix A, we can choose

[line; P2B5, P2B7]

the first row in A as the pivot row, and the element 1 in its

[circle; P2B5, P2B5]

first column is the first pivot since it is non-zero.

[write; Step 2: Eliminate entries below as 0, P3B3]

Step 2: Eliminate elements below in this column as 0

Then, we need to eliminate entries below the pivot in column 1,

[line; P2C5, P2C7]

For row two, the element of its first column is

[circle; P2C5, P2C5]

2,

This row can be eliminated by subtracting 2 times row 1

[write; Updated Row2: Row2 - 2 * Row1:, P3D3]

[write; $\text{\text{(2, 5, 5) - 2*(1, 2, 4)}$, P3E4]

[write; $\text{= (2 - 2*1, 5 - 2*2, 5 - 2*4)}$, P3F3]

[write; = (0, 1, -3) , P3G3]

Its component-wise arithmetic can be calculated by:

[line; P2D5, P2D7]

And for row three, the elements of its first column is 2,

[circle; P2D5, P2D5]

This row can be eliminated by subtracting 2 times row 1

[write; Updated Row3: Row3 - 2 * Row1, P3I3]

Its component-wise arithmetic can be calculated by:

[write; $\text{\text{(2, 1, 12) - 2*(1, 2, 4)}$, P3J4]

[write; $\text{= (2 - 2*1, 1 - 2*2, 12 - 2*4)}$, P3K3]

[write; = (0, -3, 4) , P3L3]

So the Matrix B is the result after first-column elimination:

[write; $B = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -3 \\ -3 & 4 & \end{bmatrix}$, P3N2]

we label it as Matrix B.

[break]

OK, we already know how to perform Gaussian elimination in column 1.

Applying it to column 2, that is

[write; Step 3: Move down: go to the next column, P4B3]

Step 3: Move down: go to the next column,

For column 2, the second pivot can be the second row,

[circle; P205, P205]

second column in the Matrix B, that is 1

Use the second pivot to eliminate

[circle; P2P5, P2P5]

the elements below in Column 2:

[line; P204, P206]

[line; P2P4, P2P6]

The third row can be eliminated by adding 3 times row 2

[write; Updated Row3: Row3 + 3 * Row2, P4D3]

Its updated component can be calculated by:

[write; \text{((0, -3, 4) + 3*(0, 1, -3))}, P4E4]

[write; \text{=(0 + 3*0, -3 + 3*1, 4 + 3*-3)}, P4F3]

[write; \text{=(0, 0, -5)}, P4G3]

Finally, we get the REF representation of Matrix A after eliminating all the columns

[write; REF(A)=\begin{bmatrix} 1 & 2 & 4 & \ 0 & 1 & -3 \\ 0 & 0 & -5 \end{bmatrix}, P4I3]

We can find the number of its nonzero rows is three.

[line; P4I6, P4I6]

[line; P4J7, P4J7]

[line; P4K8, P4K8]

Therefore:

[write; rank(A)=3, P4L4]

The rank of A is 3

[break]

B.5 Matrix Rank Guidance

We'll now guide step by step on the new matrix A.

[write; A=\begin{bmatrix} 1 & 2 & 3 & \ 2 & 1 & 4 & \ 3 & 3 & 7 \end{bmatrix}, P1B3]

[write; Step 1: Identify the first pivot of matrix A , P1E3]

[write; answer: the first pivot is 1.]

[break]

Great! Use the first pivot to clear column 1.

[write; Step 2: Show the updated Row2 and Row3, Pxx3]

[write; answer: updated Row2=[2 1 4]-2[1 2 3]=[0,-3,-2]; updated Row3=[3 3 7]-3[1 2 3]=[0,-3,-2]]

[break]

[write; Updated Row2=(0 -3 -2), Pxx3]

[write; Updated Row3=(0 -3 -2), Pxx3]

Great! You correctly identified the updated row 2 and row 3.

So the Matrix after clearing column 1 is

[write; \begin{bmatrix} 1 & 2 & 3 & \ 0 & -3 & -2 & \ 0 & -3 & -2 \end{bmatrix}, Pxx3]

Now use the second pivot to eliminate Column 2.

[write; Step 3: Show the 2nd round updated Row 3., Pxx3]

[write; answer: updated Row3=[0 -3 -2]-[0 -3 -2]=[0,0,0]]

[break]

[write; Updated Row3=(0 0 0), Pxx3]

Great! You correctly identified the updated row 3.

So let's write down the REF and conclude the rank.

[write; Step 4: What is the REF(A) and Rank(A)?, Pxx3]

[write; answer: REF(A)=\begin{bmatrix} 1 & 2 & 3 & \ 0 & -3 & -2 & \ 0 & 0 & 0 \end{bmatrix}, rank(A)=2]

[break]

B.6 Matrix Rank Practice

Compute the rank of the matrix A using Gaussian elimination.

[write; Compute the rank of the matrix A using Gaussian elimination, P1B3]

[write; A=\begin{bmatrix} 1 & 2 & 1 & \ 0 & 1 & 3 & \ 2 & 5 & 4 \end{bmatrix}, P1D3]

[write; answer:

Use the first pivot to clear Row 3:

Updated Row3=Row3-2*Row1=[2,5,4]-2*[1,2,1] = [2-2*1, 5-2*2, 4-2*1] = [0,1,2]

Use the second pivot to clear Row 3:

Updated Row3=Row3-Row2=[0,1,2]-[0,1,3] = [0-0, 1-1, 2-3] = [0,0,-1]

REF and conclusion:

REF(A)= \begin{bmatrix} 1 & 2 & 1 & \ 0 & 1 & 3 & \ 0 & 0 & -1 \end{bmatrix}

Rank(A)=3

]

[break]